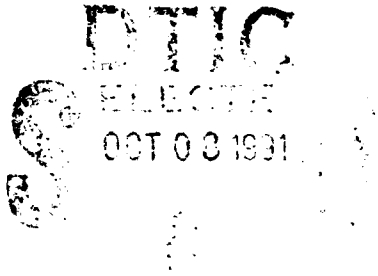
RL-TR-91-218
Final Technical Report
September 1991

ADAPTIVE NATURAL LANGUAGE PROCESSING

BBN Systems and Technologies



Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. 7302

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-218 has been reviewed and is approved for publication.

APPROVED: Douglas A. White

DOUGLAS A. WHITE
Project Engineer

FOR THE COMMANDER: Raymond P. Urtz

RAYMOND P. URTZ, JR.
Technical Director
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(C3CA) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1991	Final Mar 90 - Mar 91

4. TITLE AND SUBTITLE
ADAPTIVE NATURAL LANGUAGE PROCESSING

5. FUNDING NUMBERS
C - F30602-87-D-0093
Task: 8
PE - 61101E
PR - G302
TA - QA
WU - 01

6. AUTHOR(S) Damaris Ayuso, Sean Boisen, Robert Bobrow, Herbert Gish, Robert Ingria, Marie Meteer, Jeff Palmucci, Richard Schwartz, Ralph Weischedel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BBN Systems and Technologies
10 Moulton Street
Cambridge MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington VA 22209

Rome Laboratory (C3CA)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-91-218

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Douglas A. White/C3CA/(315) 330-3564

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A handful of special purpose systems have been successfully deployed to extract pre-specified kinds of data from text. The limitation to widespread deployment of such systems is their assumption of a large volume of handcrafted, domain-dependent, and language-dependent knowledge in the form of rules.

A new approach is to add automatically trainable probabilistic language models to linguistically based analysis. This offers several potential advantages: 1) Trainability by finding patterns in a large corpus, rather than handcrafting such patterns. 2) Improvability be re-estimating probabilities based on a user marking correct and incorrect output on a test set. 3) More accurate selection among interpretations when more than one is produced. 4) Robustness by finding the most likely partial interpretation when no complete interpretation can be found.

14. SUBJECT TERMS
Natural Language Understanding, Message Understanding, Probabilistic Modeling

15. NUMBER OF PAGES
84

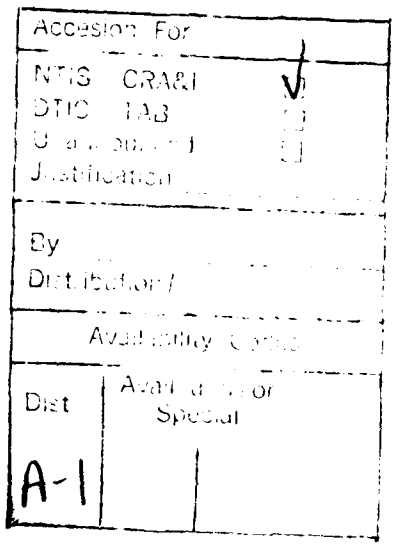
16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	U/L

ACKNOWLEDGEMENTS

The work reported here was supported by the Advanced Research Projects Agency and was monitored by the Rome Air Development Center under Contract No. F30602-87-D-0093. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

We wish to acknowledge the many contributions of Lance Ramshaw during the first quarter of this project.



EXECUTIVE SUMMARY

The terms *data extraction from text* and *data base generation* have been used synonymously to refer to the problem of automatic update of a pre-specified, formatted data base from a stream of natural language messages. A handful of special purpose systems have been successfully deployed to extract pre-specified kinds of data from text. The limitation to widespread deployment of such systems is their assumption of a large volume of handcrafted, domain-dependent, and language-dependent knowledge in the form of rules. Moving to a new topic or to a new application domain may require as much work for the second domain as for the first, since there is little that carries over from one domain to the next.

One of the critical problems in intelligent processing of natural language is the determination of the interpretation of a piece of text or spoken language. The traditional approach is the use of handcrafted linguistic knowledge (such as a grammar stating how words can combine to form meaningful units) and handcrafted domain knowledge (e.g., military units can be deployed to locations) to determine what is literally meant by a statement. The performance of such systems is hindered by the following two complementary problems:

1) Frequently more than one interpretation remains even after all linguistic and domain knowledge has been used in processing an input.
2) Partial interpretation, when no complete interpretation can be found, is difficult or impossible.

A new approach is to add automatically trainable probabilistic language models to linguistically based analysis. This offers several potential advantages:

1) Rapid development of domain-dependent and language-dependent data by finding patterns in a large corpus, rather than handcrafting such patterns.
2) Improvability by re-estimating probabilities based on a user marking correct and incorrect output on a test set.
3) More accurate selection among interpretations when more than one is produced.
4) Robustness by finding the most likely partial interpretation when no complete interpretation can be found.

This twelve-month effort is a pilot study to explore the feasibility of marrying statistical techniques to linguistically motivated technology. The three primary measures of the effectiveness of the algorithms are their reliability in handling unknown words, their reliability in assigning the correct (syntactic) form to sentences, and their ability to assist in the classification of text into relevant topics.

Several of our results are summarized as follows:

Reliability in Handling Unknown Words

- **We achieved a five-fold reduction in error rate in predicting the part of speech of unknown words.** In processing unknown words, the best error rate on predicted part of speech as reported in the literature is 75%. We were able to reduce the error rate for unknown words to 15%.

- **We demonstrated that probability models can improve the performance of knowledge-based syntactic and semantic processing.** Adding a context-free probability model improved unification predictions of syntactic and semantic properties of an unknown word, reducing the error rate by a factor of two compared to no model.

- **Much less training data than theoretically required proved adequate.** As little as 64,000 words of supervised training data was used; with 1,000,000 words of supervised training, less than a 1% improvement in error rate resulted.

Reliability in Assigning Syntactic Form

- **We obtained a reduction in error rate in**

selecting the correct interpretation of a sentence by a factor of two compared to no model. A context-free probability model on supervised training of only 80 sentences was used in this experiment.

Classifying Text Into Relevant Topics

- **A simple classification algorithm proved quite effective in detecting relevant versus irrelevant articles.** In the following results, "recalled" is the probability that a message in the class would be classified correctly, and "filtered" is the probability that a message not in the class would be classified correctly.

Category	Recalled	Filtered
BOMBING	100% recalled,	83% filtered
MURDER	87% recalled,	53% filtered
KIDNAP	76% recalled,	93% filtered
ARSON	97% recalled,	97% filtered

Our pilot experiments indicate that our new approach to text processing is both feasible and promising.

One of our most innovative results is the automatic induction of semantic knowledge from annotated examples; the use of probabilistic models offers the induction procedure a decision criterion for making generalizations from the corpus of examples.

1. INTRODUCTION

1.1 Purpose of This Report

This paper reports the results, both positive and negative, of a twelve month pilot study on data extraction from text.

1.2 The Problems in General

In order to meet the information processing demands of the next decade, natural language systems must have the capability of processing very large amounts of text, commonly called "messages", from highly diverse sources written in any of a few dozen languages. One of the key issues in building systems with this scale of competence is handling large numbers of different words and word senses. Natural language understanding systems today are typically limited to vocabularies of less than 10,000 words; tomorrow's systems will need vocabularies at least 5 times that to effectively handle the volume and diversity of messages needing to be processed.

One method of handling large vocabularies is simply increasing the size of the lexicon. Research efforts at IBM [Chodorow, et al. 1988; Neff, et al. 1989], Bell Labs [Church, et al. 1989], New Mexico State University [Wilks 1987], BBN [Crowther 1989] and elsewhere have used mechanical processing of on-line dictionaries to infer at least minimal syntactic and semantic information from dictionary definitions. However, even assuming a very large lexicon already exists, it can never be complete. Systems aiming for coverage of unrestricted language in broad domains must continually deal with new words and novel word senses.

Systems will have the additional problems of an exploding search space, of disambiguating multiple syntactic and semantic possibilities when full interpretations are possible, and of combining partial interpretations into something meaningful when a full interpretation is not found. For instance, in *The Wall Street Journal*, the average sentence length is 21 words. In a set of messages from the Foreign Broadcast Information Service, the average sentence length is 28 words, more than twice the average sentence length of the corpus for the Air Travel Information System used in the DARPA Spoken Language Systems research. If the worst case complexity of a parser is n^3, then the search space can be eight times worse than in spoken language interfaces.

Perhaps the most critical technical challenge to widespread applicability of existing natural language technology is its dependence on handcrafting rules (knowledge) at all levels of processing. Automatic acquisition of such rules is critical to reducing the cost of applying the technology to a given application domain.

A key element of our approach to these problems is the use of probabilistic models to control the greatly increased search space and to automatically acquire required knowledge from example text. We have observed that the state of the art in natural language processing (NLP) today is analogous to that in speech processing roughly prior to 1980, when purely knowledge-based approaches required much detailed, hand-crafted knowledge from several sources (e.g., acoustic, phonetic, etc.). Speech systems then, like NLP systems today, were brittle, required much hand-crafting, were limited in accuracy, and were not scalable. A revolution in speech technology has occurred since 1980, when probabilistic models were incorporated into the control structure for combining multiple sources of knowledge (providing improved accuracy and increased scalability) and as algorithms for training the system on large bodies ("corpora") of data were applied (providing reduced cost in moving the technology to a new application domain).

1.3 A New Approach

In our approach, we employ probabilistic models at all levels of processing.

Probabilistic modelling offers the following:

- High performance in template fill, since our statistical approach provides best-fit pattern-matching rather than the more rigid pattern-matching of today's knowledge-based techniques.
- Trainability from statistical analysis over large corpora, rather than having to build all rules and all knowledge by hand.
- Improvability, since feedback from the user can form the basis to re-estimate probabilities.

Probability theory offers a general mathematical modelling tool for estimating how likely an event is. Probability theory can be applied at all levels of processing in data extraction, since each algorithm has an associated class of events that can be modeled.

For instance, at the morphological level an "event" can be defined to mean the occurrence of a word as a particular part of speech, e.g., past participle of a verb, a singular common noun, or a proper noun.

At the syntactic level, "event" can be defined as the use of a grammar rule. If one employs context-free rules, the probability of a particular grammatical analysis, given the sequence of lexical items identified by morphological analysis, can be approximated by the product of the probabilities of each of the rules needed in that grammatical analysis.

That is, the use of a context-free rule

$$LHS \leftarrow RHS_i$$

implies independence of the event of using some other rule.

At the level of generating templates, "event" can be defined to be the occurrence or co-occurrence of words, the occurrence of structures, the occurrence or co-occurrence of domain model elements, etc.

To employ a probabilistic algorithm one needs a training algorithm to estimate probabilities, that is, to derive probabilities from estimates of frequency of occurrence of the events of interest.

There are two kinds of training. With *supervised training*, each event in a training corpus has been marked and labelled correctly by a human. In *unsupervised training*, the training corpus has been marked and labelled by a totally automatic process, so some of the labels may be wrong but because the process is automatic, a much larger amount of data may be processed. An initial probability distribution and a set of rules defining all possible legal events are assumed; then, a procedure estimates probabilities so as to maximize the probability in the corpus. In our initial experiments in processing text [Ayuso et al., 1990], supervised training yielded better performance than unsupervised training. One issue for future research is to develop models and algorithms that can more effectively use unsupervised training.

1.4 Focus of This Pilot Study

This effort represents a pilot study which is designed to measure two concrete effects:

1. the ability to handle words outside of the lexicon, and

2. system performance in interpreting sentences, or producing partial interpretations of well formed but not completely understandable sentences.

The lack of facility of current systems in handling new words is a serious limitation in the state of the art. The techniques we propose should predict syntactic and semantic features of an unknown word or words, thereby enabling the system to automatically acquire knowledge of the new word, and to adapt its subsequent performance by making use of that knowledge.

If statistical language modelling improves system performance in determining the correct, though possibly partial, interpretation of a sentence, then statistical language modelling can impact the accuracy of language processing systems for message processing, machine translation, and spoken language systems. Thus, this approach offers a high potential payoff.

Evaluation in these two areas provides an early test of the hypothesized approach. As a consequence, we have devised a number of small experiments to test the feasibility of this new approach.

1.5 Organization of this Document

Sections 2 and 3 provide background regarding natural language processing, presenting first a summary of the state of the art with respect to data extraction from text (or message processing) and presenting second an overview of a system architecture for data extraction from text.

Sections 4 through 8 describe five classes of experiments designed to test the feasibility of this new approach. Each was designed to test the impact of probabilistic modelling on a particular component, (How each component contributes to message processing as a whole is described in Section 3).

Sections 4 through 8 are presented in the order that each component would be employed in message processing. First, a pre-process to classify text as irrelevant or relevant is described (Section 4). Second, morphological analysis, identifying the part of speech of each word, is described (Section 5). Applying probability to select among interpretations is discussed in Section 6. When no complete interpretation can be found, partial interpretations must be found (Section 7). Lastly, a means of semantic annotation is discussed as a way to bootstrap the process of acquiring the semantics of a new domain (Section 8).

Section 9 discusses general, a priori estimates of how much data will be required to train an algorithm for a particular problem.

An evaluation workshop held in February offered the potential of evaluating the impact of one of these algorithms in a complete system and setting. Our activities preparatory to this evaluation are presented in Section 10.

Section 11 includes not only the conclusions from this pilot study but also the directions we see as most promising for future work.

2. STATE OF THE ART

In this section, we review the state of the art in message processing to provide the context of our new approach. In this review, we focus on two dimensions of portability: *domain independence*, the effort required to move the natural language shell to a new domain, and *language independence*, the effort to bring the system up in a second language.

The two primary approaches to message processing, that is extracting information from open ended text, are concept-based patterns (CBP) and linguistically-based sublanguage analysis (SA).

Neither approach is adequate for the challenges of today's message processing needs, which require systems capable of handling large amounts of open text, such as a newswire, and multiple languages and domains, such as the European community.

The CBP approach is neither domain-independent nor language-independent. Changing either the language or the domain requires a completely new set of patterns; virtually nothing carries over either to a new domain or to a new language. No automatic training procedure has yet been devised. The SA approach offers the potential of domain-independence and language-independence, but has not yet proved to be so. Its biggest drawback is the fact that it is restricted to small domains rather than open-ended text.

2.1 Concept-Based Patterns

Examples of the Concept-based pattern (CBP) approach are Carnegie Group's Text Categorization System (TCS) and TRW's Fast Data Finder. TCS assumes that the user identifies all concepts of importance and the patterns for phrases that can identify those concepts. The text is matched against "patterns of words built up using arbitrary nestings of disjunction, negation, skip (up to n words), and optionality operators" [Hayes 1990].

As an example, consider a set of 100 Spanish texts on AIDS that we studied recently. A particularly effective pattern would look for the following: a country name, up to five words to be skipped, a form of the word *notificar*, up to five words, followed by a number, the word *casos*, up to five words, the acronyms HIV or SIDA, and any number of words. For the texts we explored, that rule works remarkably well for extracting numbers of cases reported. Nevertheless, failures occurred in the texts as well, for instance, when one of the words skipped was *no* (indicating no report had been filed) and when several countries were reported in the same sentence (e.g., in translation, "A and B reported n and m respectively")..

TRW's Fast Data Finder, like TCS, is a pattern-matching system formally equivalent to a finite state machine, but directly supported by hardware. SAIC and Thinking Machines Corporation similarly have finite state pattern matchers.

These commercially available systems have been deployed in a few real-world applications. TCS has been deployed to categorize items on the Reuters Newswire, and the Fast Data Finder has been deployed for filling a limited number of data base fields from open source. These have serious limitations for message processing.

One critical limitation of the CBP approach is that the set of patterns is both domain-dependent and language-dependent. For each domain and for each language of each domain, a totally new rule set must be normally created from scratch. In fact, a totally new rule set is required just to change domains even when processing the same language, e.g., English. No automatic means of building those rule sets is known.

Compounding this problem is the large number of rules required by these systems. Suppose we wrote a rule like that above for English. It would handle active sentences (e.g., *Bolivia has reported* ...); to handle passive sentences, other rules must be written

for each passive form (e.g., *123 cases of AIDS were reported...*). One would also need rules for near synonyms (e.g., *the World Health Organization was notified that .... .*, and *the World Health Organization was informed that* ....). The purpose of a grammar and a lexicon in our approach is to automatically allow for such regularities without requiring the user to write all such predictable variations.

A third limitation of the CBP approach is that those systems rely heavily on word order. While they have some success for certain applications in processing English, their suitability for a language with a more free word order, such as Japanese, is highly questionable.

2.2 Sublanguage Analysis

Linguistically-based sublanguage analysis (SA), unlike the CBP approach, has a well-defined explicit model of the morphology, syntax, and semantic properties of language. A *sublanguage* is a variant of a natural language, spoken or written, in a given domain. The earliest system employing a sublanguage model was the TAUM-METEO system [Isabelle 1984] for translating weather reports from French to English. The language used in the class of Navy tactical reports studied in the NOSC/DARPA Second Message Understanding Conference, is another example.

The key to a sublanguage is that it has stereotypical usage, in both limited syntax and limited uses of words. Consequently, sublanguages are particularly well-suited to a purely knowledge-based approach, since 1) word sense ambiguity is limited, 2) grammatical structures are constrained, and 3) for small domains, the closed world assumption holds (i.e., one can pre-program most required knowledge). Unisys's Pundit [Hirschman, et al., 1989] and NYU's Proteus [Grishman, et al., 1989] are examples of the knowledge-based, handcrafted sublanguage approach.

Sublanguages, however, are very limited in the class of messages that can be represented. Handling open-ended domains like news articles and technical papers, if doable at all, would require substantial improvements in the breadth of language covered, as well as dramatic increases in the amount of handcrafted knowledge required. Furthermore, purely knowledge-based approaches tend to be brittle rather than robust; they would require a breakthrough in portability and scalability.

2.3 Hybrid Approaches

There are approaches that are hybrids of CBP and SA. For instance, GE's SCISOR [Jacobs 1990] uses CBP techniques to identify which sentences to focus on. The first phase of processing seems akin to concept spotting via a set of hand-coded, finite state, semantic grammars. Using at least a partial grammar of English, phrases are then syntactically and semantically identified. Then ad hoc "meta-rules" state how to combine the identified phrases into templates corresponding to relationships among entities. The meta-rules are knowledge-based constraints specific to each language and domain with ad hoc, hand-built preferences to understand sentences that are judged important.

Cognitive Systems Inc. has a commercial product, ATRANS, for processing interbank telexes of international money transfers. A mix of concept-based patterns (to find basic entities and relations) and knowledge-based techniques (to infer relations among entities) is employed. The product is specific to English interbank telexes. No domain independence nor language independence is claimed or supported.

In summary, the hybrid approaches will require substantial effort by the system builders to be ported to each new language and each new domain.

Their reliance on handcrafted heuristics and knowledge specific to each domain and language means that, even if successfully ported to a different language,

1) the same effort would be required for each new language and domain combination and

2) natural language experts who understand the components of the system would still be required for a new language and domain pair.

Knowledge acquisition is critical to each domain and language, but is not automated nor based on data-driven training. While the hybrid approach is less brittle than most by having some generalization rules, there is no mathematically-based training mechanism for ranking alternatives; ad hoc preferences govern how the search progresses.

2.4 Conclusion

Though a handful of systems have been successfully deployed, both the state of the art as deployed and the state of the art as represented in laboratory systems face a challenge. The effort to port natural language system to a new domain or to a new language is perhaps the most serious roadblock to further deployment of these systems.

Probabilistic models offer a new approach, automatic acquisition of the required knowledge from example text. Additionally, probabilistic models can supplement the state of the art by less brittle, more accurate algorithms than current knowledge-based algorithms.

3. MESSAGE PROCESSING SYSTEM ARCHITECTURE

In this section we provide an overview of our approach to message processing by laying out in detail our system architecture. The modularization shown in the diagram Figure 3-1 reflects two underlying themes of our approach:

1) To identify and isolate each process and intermediate representation so that we can define a linguistically motivated set of "events" that can be most effectively used by the probabilistic models

2) To isolate the various knowledge sources that the system needs so that the parts that must be language dependent or domain dependent (such as the lexicon) are separate from more general knowledge sources, and furthermore, to isolate these knowledge sources from the domain/language algorithms that operate over them.

This modularization not only makes the system more portable, but it also lets us experiment with different kinds of processing algorithms and different forms for the knowledge sources. This is a key point since the goal of the project is to explore a range of ideas, not simply build a single system.

In Figure 3-1, hexagons represent dynamic data structures used by the system in processing the input. The flow of control is represented by the broad downward arrows through the processing boxes (rectangles). This is a pipeline flow, rather than a strictly sequential flow, that is, one process need not complete a message before the next process begins. The hexagons are dynamic data structures that are created and manipulated in the course of processing, in contrast to the knowledge structures depicted by ovals on the right hand side of the diagram, which are static during processing.

The data structures are in principle open to inspection by any process, though as shown in the diagram they function mainly to mediate between two processes. Keeping them independent of any particular process will allow us to later investigate the advantages of a more open architecture that incorporates parallelism (e.g. continue morphological processing while the previous sentence is being processed by the parser) and communication from later processes back to earlier processes (e.g. the confirmation of a referring expression in the discourse module could be used to constrain the parsing component).

In addition to the processing components illustrated, we have built a preprocessor which classifies text according to its relevance and topic (described in chapter 4). This component will allow the system to ignore paragraphs that are irrelevant and focus on those that contain relevant information, greatly increasing the efficiency of the overall system.

Furthermore, the diagram does not show the acquisition, training, and editing components that are used to create the (oval) knowledge bases and probability estimates.

3.1 Control Flow

The input to the system is an entire message. The morphological processor finds words, punctuation, headers, etc. in the input and determines sentence boundaries. In addition, the process also marks part of speech and other morpho-syntactic information available from the lexical items.

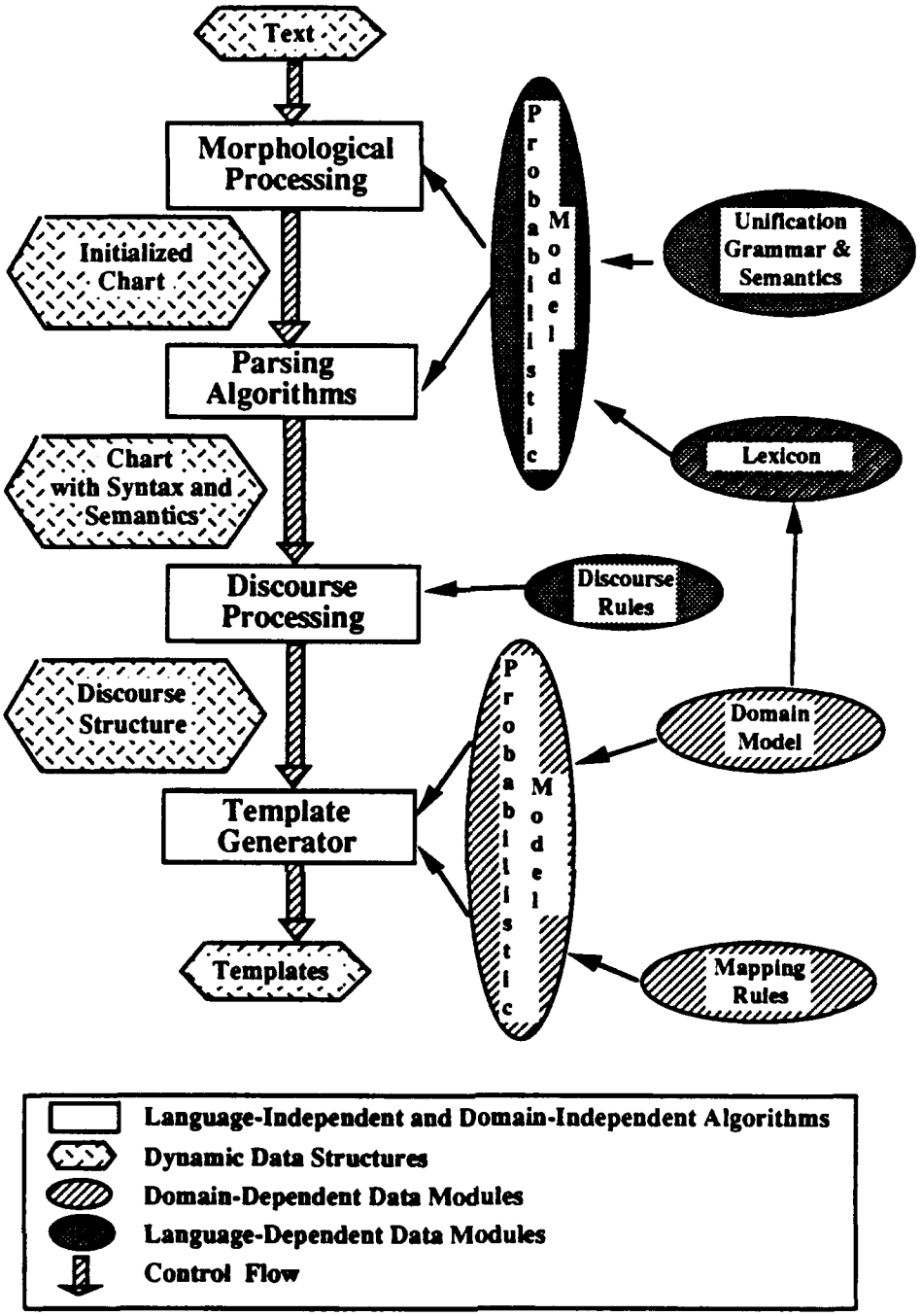


Figure 3-1 System Architecture

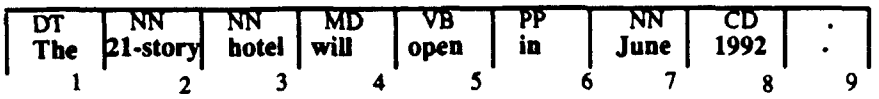


Figure 3-2 Initialized Chart

The morphological component then initializes the chart, as shown in Figure 3-2. The chart is a well known data structure capable of representing weighted alternative sequences of entities (initially, the entities are symbols, words, or punctuation; the parser augments this to include parsed phrases).

Each edge of the chart contains: the entity represented by the edge (character, word, punctuation, phrase, etc.), the segment of original input spanned by the edge, and, if applicable, part of speech, root form, and features (from the dictionary or from morphological analysis). Every entry in the chart can be assigned a probability, though for simplicity we have not shown that in Figure 3.2.

The parsing component then uses the chart to build the syntactic structure of the text, extending and modifying it as new levels of structure are found. In our current approach, we use the MIT Fast Parser, which generates fragment parses spanning the input (see Section 7.4).

The next level of processing is the semantic interpreter, which operates on the parse fragments produced by the parser, and assigns them fragment semantics. The interpreter is discussed in the next section.

The discourse component operates over each sentence as the semantics for that sentence is produced; however, the structure it builds stays active for the entire processing of the message and in the end spans the entire text. In contrast, the chart is reinitialized by the morphological component for each sentence. The discourse component is described in section 3.3.

The final process is template generation, which uses the discourse structure to fill the templates. This process does not run until the entire discourse structure for the message has been built. Waiting until the entire message has been processed avoids false starts, such as when the introductory sentences imply a date, but there are several dates of interest in the text. The template generator is described in the final section of this chapter.

A structured representation of the processed message is the end result, where a tree structure connects all components of the message. The created events and templates are attached at a high level. This is illustrated in figure 3-3.

3.2 Semantic Interpreter

The semantic interpreter operates in a bottom-up, compositional fashion. Throughout the system, defaults are provided so that missing semantic information or rules don't produce errors, but simply mark semantic elements as unknown. This is consistent with our belief that partial understanding has to be a key element of text processing systems, and missing data has to be regarded as a normal event rather than a system error.

The semantic rules are based on general syntactic patterns, using wildcards and similar mechanisms to provide an extra measure of robustness. The basic elements of our semantic representation are "sem-forms", each of which introduces a variable with a type taken from the domain model, and a collection of predicates pertaining to that variable. The semantic types represented include events, entities, and states-of-affairs, each of which can be known, referential, or unknown.

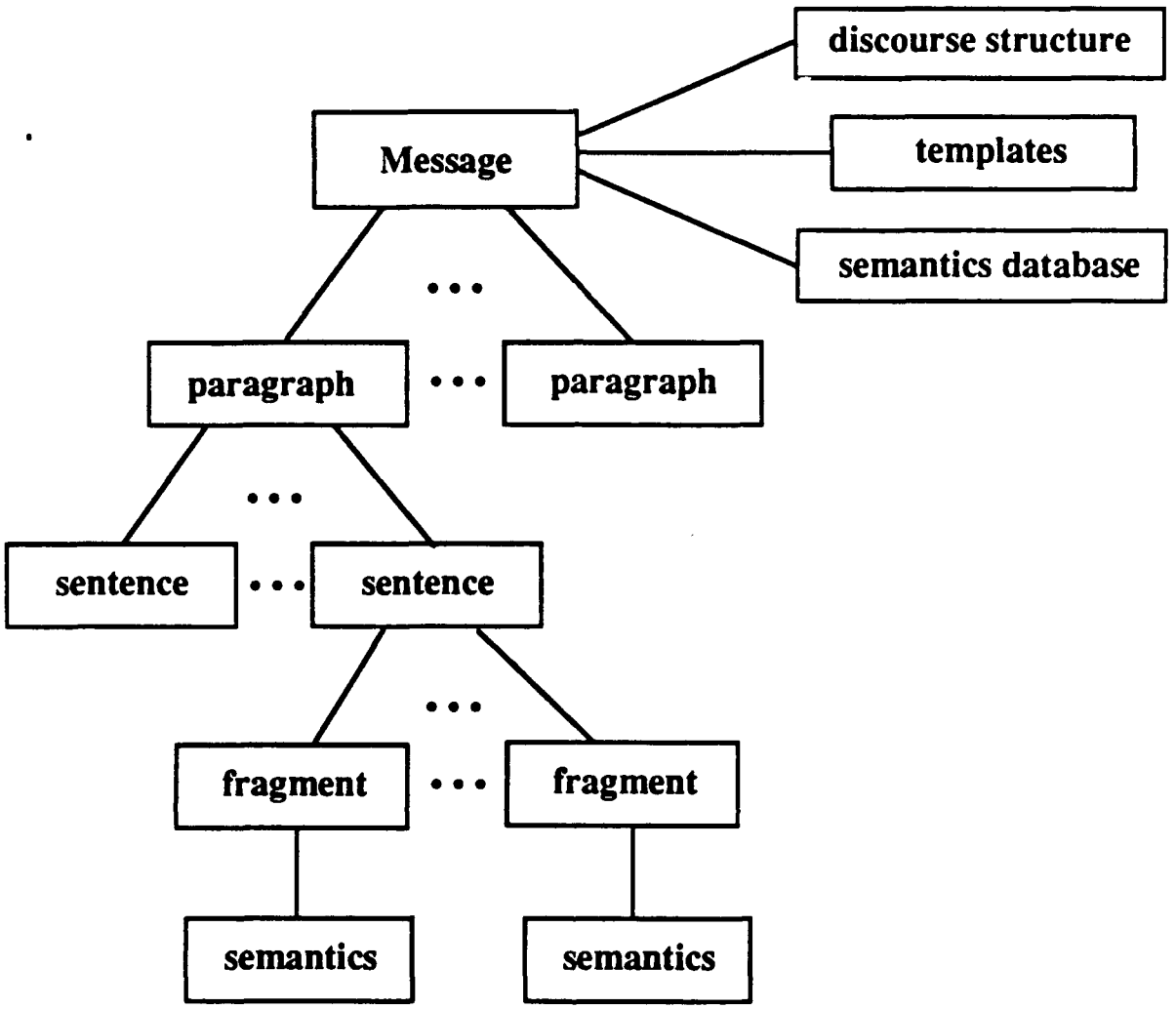


Figure 3-3: A Message Structure

As an example, take this sentences from message 0001 in the MUC-3 development corpus.

THE ARCE BATTALION COMMAND HAS REPORTED THAT ABOUT 50 PEASANTS OF VARIOUS AGES HAVE BEEN KIDNAPPED BY TERRORISTS OF THE FARABUNDO MARTI NATIONAL LIBERATION FRONT [FMLN] IN SAN MIGUEL DEPARTMENT.

The MIT Fast Parser produces six trees for this (three of them consisting solely of punctuation). Here is the first:

```
(S (NP (DETERMINER "THE")
       (ADJP (ADJ "ARCE"))
       (N "BATTALION")
       (N "COMMAND"))
   (VP (AUX (V "HAS"))
       (VP (V "REPORTED")
```

```
(S (COMP "THAT")
   (S
    (NP
      (NP (DETERMINER (ADV "ABOUT")
            (DETERMINER (NUM "50")))
          (N "PEASANTS"))
      (PP (PREP "OF")
          (NP (ADJP (ADJ "VARIOUS"))
          (N "AGES"))))
  (VP
     (AUX (V "HAVE")
          (V "BEEN")
          (PP (PREP "BY")
              (NP (NP (N "TERRORISTS"))
                (PP (PREP "OF")
                 (NP (DETERMINER "THE")
                     (N "FARABUNDO"
                        "MARTI"
                        "NATIONAL"
                        "LIBERATION"
                        "FRONT"))))))
      (VP (V "KIDNAPPED") (NP)))))))
```

In our semantic representation, there are three basic classes: entities of the domain, events, and states of affairs (SOA). Entities correspond to the people, places, things, and time intervals of the domain. These are related in important ways, such as events (who did what to whom) and states of affairs (properties of the entities). Entity descriptions typically arise from noun phrases; events and states of affairs may be described in clauses. Variables in our semantic representations appear as a number preceded by a question mark.

Here is the semantic representation for the first tree (some details are omitted to clarify the exposition: the full form can be found in the appendix):

```
(?38
   ((KNOWN-EVENT ?38 COMMUNICATION
              (AGENT-OF ?38 ?4)
              (OBJECT-OF ?38 ?35))
    (KNOWN-ENTITY ?4 PEOPLE
              (SOCIAL-ROLE-OF ?4 MILITARY)
              (DESCRIPTION-OF ?4 "THE ARCE
                BATTALION COMMAND"))
    (KNOWN-EVENT ?35 KIDNAPPING
              (OBJECT-OF ?35 ?13)
              (AGENT-OF ?35 ?29))
    (KNOWN-ENTITY ?13 PERSON
              (PP-MODIFIER ?13 ?12 "OF")
              (SOCIAL-ROLE-OF ?13 CIVILIAN)
              (NUMBER-OF ?13 50)
              (DESCRIPTION-OF ?13 "ABOUT 50
                PEASANTS"))
    (KNOWN-SOA ?12 STATE-OF-AFFAIRS
          (NUMBER-OF ?12 PLURAL)
          (DESCRIPTION-OF ?12 "VARIOUS AGES"))
    (KNOWN-ENTITY ?23 ORGANIZATION
              (NAME-OF ?23 "FMLN")
              (SOCIAL-ROLE-OF ?23 TERRORISM)
              (DESCRIPTION-OF ?23 "THE
                FARABUNDO MARTI NATIONAL
                LIBERATION FRONT"))
    (KNOWN-ENTITY ?29 PERSON
              (PP-MODIFIER ?29 ?23 "OF")
              (SOCIAL-ROLE-OF ?29 TERRORISM)
              (NUMBER-OF ?29 PLURAL)
              (DESCRIPTION-OF ?29
                "TERRORISTS"))))
```

The main semantics here are a communication event (?38) which reports a kidnapping event (?35). The agent of the kidnapping is an undefined number of terrorists (?29), and the object is 50 civilians ("ABOUT 50 PEASANTS", ?13).

Not everything represented here has actually been understood: for example, the semantic representation of the 50 peasants (?13) includes the information that there is a prepositional phrase modifier whose preposition is OF, and whose object is the phrase "VARIOUS AGES". In this and other cases, PP-MODIFIER is used to indicate that a certain structural relation holds between these two items, even though we don't know what the actual relation is. In this instance, understanding the relation is of no consequence, since the information that the peasants varied in their ages does not contribute to the template filling task. The information is maintained so that later expectation-driven processing can find it if necessary.

The representation of the agent of the incident ("TERRORISTS", ?29) provides a good example of the value of representing incompletely-understood relationships. When a template is being generated for this, there is no name attached to the agent entity, but there is a PP-MODIFIER with OF. Because the sub-entity has a name predicate (?23, "FMLN"), we can postulate that the OF

relation here indicates membership in a (named) group, and so the proper template filler can be found.

The tail of the sentence, "IN SAN MIGUEL DEPARTMENT", is in a separate parse fragment, so the information that this is the location of the kidnapping cannot be directly recovered from the fragmentary semantics. This example points out the need for good discourse processing

3.3 Discourse Processing

The discourse component of PLUM performs the operations necessary to derive, from the semantic representation of the fragments in the input message, a high level "message event structure", or a representation of the events of interest that occurred in the message. Each event in the message event structure is similar in principle to the notion of a "frame", with its corresponding "slots" or fields. There is a correspondence between the event structure and the semantics that the semantic interpreter assigns to an event in the text. However, the semantics assigned by the interpreter can only include (at most) relations contained locally in the fragment; the discourse module must infer other long-distance or indirect relations not explicitly found by the interpreter. The template generator then uses the structures created by the discourse component to generate the final templates. Currently only terrorist incidents (and "possible terrorist incidents") generate events, since these are the only relevant events for MUC template generation.

Two primary structures are created by the discourse processor which are used by the template generator: the semantics database and the event structure.

The semantics database contains all the tuples mentioned in the semantic representation of the message. In addition, when references in the text are resolved, their variables are unified in the database. Any other inferences done by the discourse component also get added to the database. Currently there is only one database which is produced, ideally there should be several, each representing one inference path.

An EVENT structure has at least the following fields:

name : the type of event, e.g., MURDER - corresponds to a domain model concept
slots : list of slot structures
id : unique id (var) of the semantic form which gave rise to this event
triggers : the fragment(s) that gave rise to this structure
inherits-from : other event types this inherits from
criterion : a predicate which if true signals the creation of this type of event

A SLOT structure has at least the following fields:

name : slot name--corresponds to a domain model role
fill : list of fillers; each filler is a variable corresponding to a semantic form
fill-type : the expected semantic type of the filler
number: expected number of fillers
fill-test : function which returns a filler if it finds one
default: either a value or a function to determine how to fill if no filler is found initially
if-fill : a function to execute when the slot is filled
parent-event : pointer to the event structure of which this is a slot.

The following is an example of the event definition for kidnapping:

```
(Define-event KIDNAPPING
    : criterion ( find-new-event :type
                  kidnapping )
    : inherits-from TERRORIST-INCIDENT-
ON-PERSON )
```

As defined, this will cause a kidnapping event to be generated whenever the semantics shows something of type "kidnapping", which can arise, for example, from either the verb "kidnap" or the noun "kidnapping".

An example slot definition for perpetrator follows; any kidnapping event inherits this slot:

```
(Define-slot PERPETRATOR
   : parent-event TERRORIST-INCIDENT
   : fill-type PEOPLE
   : fill-test  ( find-in-database
      (PERPETRATOR-OF *event* ?x ) )
   : if-not-filled
     ( look-locally-then-globally
        ( or ( is-type? ?x TERRORIST)
           ( find-in-database
                 (SOCIAL-ROLE-OF ?x
                  TERRORISM ) ) )
```

The :fill-test indicates that if the semantic interpreter (or other inferencing procedure) has found a PERPETRATOR-OF relation already, the filler of this relation becomes the filler of the slot. The :if-not-filled procedure is a default function which will search for a possible filler, if the slot has not been filled by the end of the processing of the message.

The procedure "look-locally-then-globally" looks outward from the point in the text which triggered the event, trying to find an entity which satisfies the given predicate - in this example, it will look for any terrorist entity. This procedure will assign the filler it finds a heuristic score indicating how far from the trigger the filler was found. Currently the scores are 1 if found in the same fragment, 2 if found in the same sentence, 4 if in the same paragraph, and a score of at least 6 if found in another paragraph, incrementing the score depending on the number of paragraphs from the trigger. (In future these heuristic scores can be replaced by probabilities based on distance away.)

The discourse component must infer any relevant relation that was not found directly in the semantics of a fragment. This task includes performing reference resolution and other types of inference - all in the face of partial understanding. Currently the discourse component finds referents for simple pronouns. The additional ambiguity introduced by partial understanding is being addressed by having different "views" into the semantics database, each view representing one inference path. This work is in its preliminary stages.

Another task for the discourse component, related to the reference resolution task, is to recognize when different phrases in fact refer to the same event. Each event reference generates an event structure. Before the default filling of unfilled slots begins the discourse module attempts to merge event structures when feasible. Currently events are merged when they have the same event type and their filled slots are compatible.

As an example of the operation of this module, we continue the example begun in the previous section:

THE ARCE BATTALION COMMAND HAS REPORTED THAT ABOUT 50 PEASANTS OF VARIOUS AGES HAVE BEEN KIDNAPPED BY TERRORISTS OF THE FARABUNDO MARTI NATIONAL LIBERATION FRONT [FMLN] IN SAN MIGUEL DEPARTMENT. ACCORDING TO THAT GARRISON, THE MASS KIDNAPPING TOOK PLACE ON 30 DECEMBER IN SAN LUIS DE LA REINA

The event structure generated for this is as follows:

```
(KIDNAPPING
    (TRIGGERS (?50) (?35) )
    (TI-PERP-OF (?29 1) (?23 1) )
    (EVENT-TIME-OF (?60 1) )
    (OBJECT-OF (?13) )
    (EVENT-LOCATION-OF (?63 1) )
```

The variables which fill the slots correspond to "sem forms" in the semantics. The entities denoted by those sem forms are the real fillers of the slots. The appendix provides detailed output for this example, showing the semantics for each variable. Here we will point out the results of two of the discourse tasks mentioned previously: event

merging and the default filling of slots.

Note that the example shows two triggers for the kidnapping event. This is an indication that two events were merged - one triggered by "have been kidnapped", and the other triggered by "the mass kidnapping". The numbers that accompany some of the filler variables are the heuristic certainty scores assigned by the default - filling process. For example, the filler of EVENT-LOCATION-OF has a certainty 1, indicating it (30 DECEMBER) was found in the same fragment as one of the triggers (THE MASS KIDNAPPING).

3.4 Template Generator

The template generator takes the event structure produced by discourse processing and fills out the application-specific templates. Clearly much of this process is governed by the specific requirements of the application, considerations which have little to do with linguistic processing. For example, in our domain model, all terrorist incidents have a result: but the MUC-3 task description states that, if the incident type is MURDER, the RESULT slot is to be left unspecified. The template generator must incorporate these kinds of arbitrary constraints, as well as dealing with the basic details of formatting.

The template generator uses a combination of data-driven and expectation-driven strategies. First the information in the event structure is used to produce initial values, merging information where necessary (e.g., multiple fillers of the TI-PERPETRATOR-OF or EVENT-LOCATION-OF role). At this point, values which should be filled in but are not available in the event structure are supplied from defaults, either from the header (e.g., date and location information) or from reasonable guesses (e.g. that the perpetrator confidence is usually REPORTED AS FACT).

We expect to eventually use a classifier (as described in Section 4) at this stage of processing. This is especially appropriate for template slots with a set list of possible fillers, e.g. perpetrator confidence, category of incident, etc.

The example in the appendix has the following event structure for a paragraph in message #0001 (omitting slots which are unfilled):

```
(KIDNAPPING (?50 ?35)
            (TI-PERP-OF (?29 1) (?23 1))
            (EVENT-TIME-OF (?60 1))
            (OBJECT-OF ?13)
            (EVENT-LOCATION-OF (?63 1)))
```

This indicates a single kidnapping event which is based on two semantic events (?50 and ?35), with known information about the perpetrator, time and location of the event, and who was kidnapped.

Since two perpetrators are identified, the template generator must determine whether these are equivalent descriptions of the same entity which must be merged, or distinct descriptions. Since the MUC-3 template has slots for both identifiers of both individual and organizational perpetrators, more than one TI-PERP-OF entity might be valid: that is the case here, where ?29 represents "TERRORISTS" (which is used to fill the perpetrator individual slot) and ?23 represents the FMLN, the organization to which the terrorists belong. Similar techniques must be used throughout the template generator to deal with the problem of multiple slots, when the discourse processing is unable to determine (on the basis of linguistic information) that the fillers should be merged.

In other cases, explicit information must be merged with implicit information: for example, the time of the kidnapping event (?60, "30 DECEMBER") doesn't include the year, which must be determined from the header of the message. (Note the additional complication here that the message, dated January 3rd, was actually issued in the subsequent year, so additional logic is required to determined when an annual boundary is crossed). Likewise, the text of the article doesn't specify that San Luis de la Reina is located in the country of El Salvador; since that information is required for the template

fill, it must be supplied from the location model.

4. CLASSIFICATION EXPERIMENTS

Several possible uses of statistical text classification (the probabilistic assignment of text to categories) in a message processing system exist. For each, the goal is to provide statistically based evidence of the category of a piece of text, and incorporate this evidence into the processing of the text.

In the domain of the Third Message Understanding Conference (MUC-3), one usage is hypothesizing the template type(s), if any, that should be generated for a given article. For example, the article may mention a murder, arson, bombing, or other terrorist attack. If one can with little effort distinguish the articles that are irrelevant from those that address a category of interest, the system can process those relevant messages in greater detail, and with greater reliability. Depending on the target recall and precision desired, the system could choose to ignore articles which have been classified as probably irrelevant.

It is possible that any template field that must be filled by a fixed, small number of alternatives, may be appropriately handled by a synthesis of knowledge-based and statistical techniques.

Of these three applications, we have investigated only the first. In this chapter, we first overview types of classifier algorithms on Section 4.1, and then report our experimental results in Section 4.2.

4.1 Classification Algorithms

4.1.1 Benders Tree Classifier

As an introduction to binary tree classifiers we will consider how one goes from training data to classifier design. Consider that we have available a collection of training data where each datum is a vector of features and one entry is a class label associated with these features. For example, one entry in the vector can be a label naming the type of the controller, with the other positions denoting particular words. The entries in all but the first position of this vector would be the frequency of occurrence of the words in the body of text under consideration (whether a single utterance or an entire set of dialogues).

The first step in the design process is to split the training data into two subsets by thresholding on a single feature. That is, each feature is examined as a candidate for splitting the data at a variety of different thresholds. The feature and threshold selected as the most useful is the one which does the most to "purify" the data, where by "purify" we mean reducing the uncertainty about the class membership of the feature vectors.

An example classification tree for recognizing articles reporting arson appears in Figure 4-1. (Ovals represent leaves stating the category the text belongs in. Interior nodes in the tree represent decision criteria. If the criterion is met, move to the right child; if not, move to the left child.) The algorithm generated it based on 1,000 messages that had previously been selected by a boolean keyword search to find articles about terrorism in Latin America. These articles were then labelled by hand as to whether they reported a terrorist event of type kidnapping, bombing, murder, etc. This provided supervised training for the classifier.

Before we begin the tree growing process the only knowledge we have about the class membership of a feature vector is from the *a priori* probabilities of the class occurrences. If we have N classes and

$$P_i \quad i=1,\ldots, N$$

denote the *a priori* probabilities of the classes, then our initial uncertainty is given by the entropy of these probabilities, i.e., the entropy of the classes,

$$H(C) = \sum_{i=1}^{N} P_i \log P_i$$

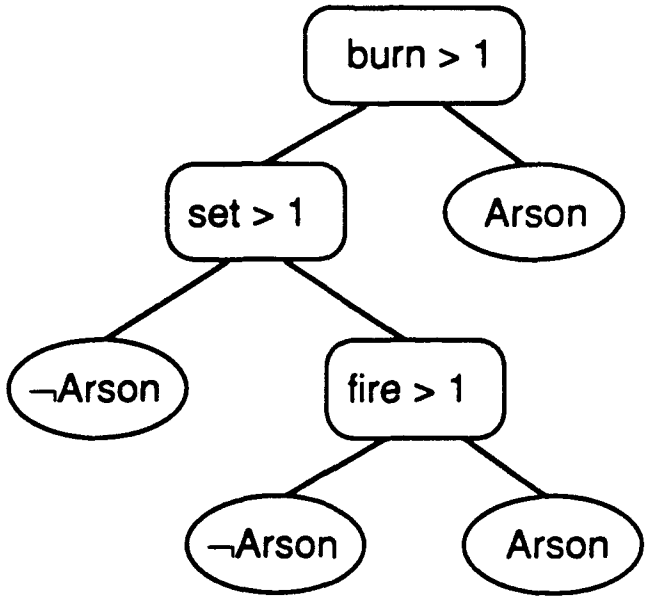


Figure 4-1 A Classification Tree for Articles for Arson

After we split the data on the value of a single feature we now have a new uncertainty of class membership. When we split the data into two groups we now have reduced our uncertainty and the new uncertainty is a conditional entropy which we will denote by

H(Cls),

which is the entropy of the classes given that the data has been split. If we let p_L denote the fraction of the data that had the feature value that was less than the threshold and

p_G

denote the fraction of data for which the feature value was greater than the threshold, we can write,

$$H(Cls) = p_L H(Cls,<t) + p_G H(Cls,>t)$$

where

<t

and

>t

denote less than and greater than than the threshold, respectively. Thus the new uncertainty is the average uncertainty, with respect to class labels, of the data on both sides of the split. The change in uncertainty obtained by splitting the data is

H(C) - H(Cls)

and represents the mutual information between the classes and the split data. As a simple example consider that there are only two classes and that the split on the feature completely separates the two classes. In this case

H(Cls)

is equal to zero since there is now no uncertainty in the classes.

After we have found the best first split point we have two data sets. We now repeat the initial splitting process again on each of the data sets. This includes allowing the same feature to split on what was used previously. Quite literally the process is repeated on each

of the subpopulations. This process is continued, at least in theory, until each leaf of the tree contains an individual member of the database, with the final tree being a pruned back version of the overgrown tree. The issue of tree pruning is quite important and has an impact on the utility of the tree classifier on new data. Designing the right size tree is another variant of the basic issue in pattern recognition of not over or under parameterizing the classifier. The method of pruning that we have followed is that developed by Breiman et. al.[CART program 1985] which relies of the use of an auxiliary data set to determine the appropriate tree size, or uses withheld data and cross validation techniques in determination of tree size.

The type of classifier we have described above has an aspect that is not typically found in other types of classifiers: it performs feature evaluation and selection as part of the classifier design. Most importantly it is finding features that are not highly correlated, i.e., it is finding features that complement one another. This is an outcome of the hierarchical nature of tree growing which requires at each level of the tree features to do new jobs: partitioning different subsets of the data conditioned of the previous values of other features. Also the greedy nature of the design algorithm ensures that a small computational penalty is exacted by the inclusion of additional features.

4.1.2 A Bayesian Alternative to CART

Although the CRT algorithm offers several benefits, and has much merit as a good first classifier to apply to a problem, it does have certain drawbacks. These drawbacks include a difficulty in finding good linear combinations of features and incorporating prior knowledge about the importance of certain information.

In addition, although the CART procedure provides a classifier, i.e., a determination of membership of each selection of text into the categories under consideration, it does not provide a scoring or ranking of class membership. Such a ranking is useful in generating ROC plots, e.g., plots of probability of correctly detecting the text as being a member of the class considered vs. the probability of falsely accepting the text as a member of the class. These plots are useful for establishing an operating point. A scoring or ranking could also be useful information to be incorporated in subsequent processing.

We have therefore developed an alternative feature evaluation procedure and classification method. The method is a basic Bayesian approach wherein the log odds of a class occurring is incremented or decremented when selected features are observed. Features (words, phrases, and equivalence classes of words and phrases) are selected on the basis of their frequency of occurrence, and their log odds, which are measured from the training corpus.

This approach is flexible and provides a score, the accumulated log odds, that can be used to rank text with regard to class membership. The method can also be viewed as generating a weighted combination of the features for scoring the text. Prior information can be readily incorporated in this approach. At present the assumption is made that features are statistically independent, if need be it is possible to modify this classifier to include feature dependencies.

4.2 Experiments in Classification

One classifier (described in section 4.1.2) was used in experiments to distinguish murder and non-murder articles. For each article, it accumulates the empirical odds for each word root, i.e., the ratio of a particular word root's count in the murder articles to its count in non-murder articles. By selecting threshoids over which an article is to be considered a murder article, we can plot the probability of detection *pd* (i.e., the probability that a murder article will be correctly classified, or *recall*, versus the probability of false detection *pf* (i.e., the probability that a non-murder article will be classified as a murder article). The resulting plot is the ROC curve in the figure below.

Given that study showing the tradeoff in

probability of detection versus probability of false detection, we explored using the CART classification algorithm in predicting the type of template(s) to be generated from an incoming message.

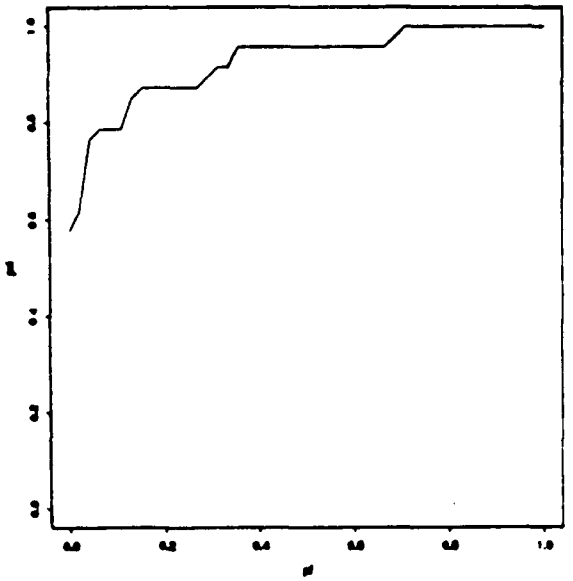


Figure 4-1 ROC Plot: Murder vs. non-Murder Articles.

Our first experiments were not fully successful, but they led us to an effective classification strategy. The features used in our initial experiment (with 42 articles as training) were simply word roots. A vector corresponding to an article contained word counts for each root word found in the article. Counts were done for all nouns and verbs. In studying the vectors generated, we found discriminating words such as "attack", "assassination", and , "kidnap", which occurred much more often in the relevant articles than in the irrelevant ones. However, given the disjointness of topics covered in the relevant articles, CART was unsuccessful in doing the classification with so few examples in the training set.

Those first disappointing results in using CART to classify articles directly into the classes RELEVANT and IRRELEVANT led us to classify into the more specific classes IRRELEVANT, MURDER, BOMBING, and KIDNAPPING. We used 106 "pure" articles (those only belonging to one of those classes) for training out of the 142 articles we had at that point. Furthermore, we grouped the word features together by stem, so that, for example, "kidnapper" and "kidnap" were grouped together in one feature. We used cross-validation to maximize the data size (where the program would repeatedly train on 90% of the data and test on 10%, until all the data is tested on). For the two categories making up most of the data, IRRELEVANT and MURDER, CART correctly classified irrelevant articles 85% of the time, and murder articles 76% of the time on the training set as summarized in Figures 4-2 and 4-3.

Although cross validation is quite fair in estimating error rates in most domains, its accuracy has been disputed in word-based classifications. Crawford et al. argue that cross-validation is based on the assumption that the training set includes samples fairly uniformly distributed over the possible domain, and that a small set of articles does not adequately represent the domain of all English messages.

The original set of approximately 150 messages with answers for the Third Message Understanding Conference (MUC-3) still appeared to be insufficient training data for the classification algorithm The arrival of the full 1300 MUC messages provided us with a much larger message set for classification. Given such a large set of messages, in December we used 1000 articles as the training set, and 300 as the test set. Again, training vectors were generated based on word stem counts.

Class 1 = IRRELEVANT
Class 2 = MURDER
Class 3 = BOMBING
Class 4 = KIDNAPPING

		TRUE CLASS			
		1	2	3	4
PC	1	0.85	0.24	0.33	1.00
r l	2	0.13	0.76	0.08	0.00
e a	3	0.02	0.00	0.58	0.00
d s	4	0.00	0.00	0.00	0.00
i s					
c					
t					
e					
d					

Figure 4-2. LEARNING SAMPLE CLASSIFICATION PROBABILITY MATRIX

		TRUE CLASS			
		1	2	3	4
PC	1	0.72	0.24	0.42	0.50
r l	2	0.26	0.71	0.08	0.50
e a	3	0.02	0.04	0.50	0.00
d s	4	0.00	0.00	0.00	0.00
i s					
c					
t					
e					
d					

Figure 4-3. CROSS VALIDATION CLASSIFICATION PROBABILITY MATRIX

One set of experiments was classification of articles into a specific subcategory or not. We trained CART to generate a boolean classification tree for each relevant subcategory, that is, a tree which says whether the article is or is not in the category. Given an article, the different classification trees can be applied to it to determine which relevant categories the article is related to. In the results in figure 4-4, "recalled" is the probability that a message in the class would be classified correctly, and "filtered" is the probability that a message not in the class would be classified correctly.

BOMBING	100% recalled, 83% filtered
MURDER	87% recalled, 53% filtered
KIDNAP	67% recalled, 93% filtered
ARSON	97% recalled, 97% filtered

Figure 4-4 Results of First CART Classification Experiment

It is interesting to note that cross validation did indeed give more optimistic error rates than those found with the test set.

A second attempt at direct classification of subclasses was geared to both incorporate the newly available data, and to utilize a higher percentage of the messages. Previous tests used only pure articles, i.e., those containing only one subcategory (e.g., murder, but not murder *and* kidnapping). About 1/3 of the message set is pure.

For this experiment, we generated a training vector for each subclass in which the message was a member. This gave us a less precise, but much larger, training set. Finally, misclassification costs between relevant categories were cut, so that the imprecision we had introduced would not lead CART away from the primary goal of partitioning relevant from irrelevant. On the 300 message test, the classification tree was able to recall 86% of the relevant messages and correctly reject 39% of the irrelevant messages.

Although these were good results, we plan to experiment further with this technique. The increase in the number of pure articles could have very well pushed us beyond the point where we would need to scavenge for data this way. We have yet to run a test to see if this is true. Furthermore, because of the large amount of work involved in hand classifying 1300 messages, it would be interesting to cut down our training set to see how much the scavenging improves performance given less information.

4.3 Future Work

A second use of text classification in our message processor is to apply the classification techniques just discussed, but at the paragraph level, to assign categories to individual paragraphs. One of the uses of paragraph-level classification is to indicate possible shifts in topic, suggesting that more than one template type may need to be generated for an article. Other advantages are analogous to those mentioned above for articles, such as being able to ignore, or at least to devote less processing to, paragraphs classified as irrelevant.

For example, given a new article which is considered relevant, the system would first perform a preliminary fast parse over the article, identifying the noun phrases present, their semantic classes, and recording the overall structure of the text. Including irrelevant paragraphs in this first pass is necessary for finding global information in the article, such as dates and locations, that appear in paragraphs labeled irrelevant. A second, more detailed pass, would process only those paragraphs classified as relevant. The category assigned to a paragraph by the classifier would determine its template type in case of ambiguity.

A third use of text classification pertains to the filling of template fields that range over a fixed set. For example, YES/NO fields such as the PROTOTYPE field of the MICROELECTRONICS FABRICATION template, or set fields such as the BUSINESS CODE field of the COMPANY templates in the TIPSTER domain. The same techniques described above may be used to classify articles (or smaller text segments) into such fixed sets. One may expect that when there is not enough evidence in the understood text, using the filler indicated by the classifier would provide substantially better than chance performance. Assuming the system also indicates the degree of certainty in assigning fillers, this would be a valid and useful application.

The preliminary classification experiments described in this section of the paper were based on occurrences of word roots. Given the ambiguity of words in isolation, one would expect better classification performance using higher-level features such as the semantic categories of phrases. In an experiment under the DARPA/RADC Gisting program, we found that the use of higher-level features improved performance in a separate application of CART, which classifies air traffic control conversations. Based on this result, in this project, we plan to use quick parsing techniques to facilitate the extraction of such features, if it proves to pay off in terms of speed, accuracy, etc. We will also experiment with different types of classifiers to determine the best type for each kind of classification task.

5. PART OF SPEECH LABELLING

Unknown words and novel uses of known words arise unavoidably in sources of open-ended text, such as a newswire. A system should be able to tell the part of speech of a new word, for instance, that it is a verb (stating an action or state of affairs), that it is a common noun (stating a class of persons, places, or things), that it is a proper noun (naming a particular person, place, or thing), etc. If it can do that well, then more precise classification and understanding is feasible. For known words, determining the part of speech and in context is the first step in interpreting the word. Using probability models to reduce the number of alternatives early on can greatly reduce the search for an interpretation of a sentence or a message.

It is straightforward to predict the part of speech of a word using probabilities.

Many words are ambiguous in several ways such as the following:

a *round* table: adjective
a *round* of cheese: noun
to *round* out your interests: verb
to work the year *round*: adverb

Even in context, part of speech can be ambiguous, as in the famous example "Time flies," where both words could be either a noun or a verb. However, the interpretation where "time" is a noun and "flies" is a verb is more likely than the other three.

Models predicting part of speech can serve to cut down the search space a parser must consider in processing known words and can be used as one input to more complex strategies for inferring lexical and semantic information about unknown words. We have explored the use of such models in both contexts using our part of speech tagger, POST.

5.1 Bi-gram, tri-gram, n-gram models

We can construct a simple, yet powerful statistical model of language by measuring the likelihood of short sequences of words or word classes. This is generally called the *n-gram* model of language. For example, knowing one word in a sentence, we can make reasonable predictions about the following word. This model is commonly used in speech recognition when we want to choose the sentence (sequence of words) that is most likely, given the acoustic model and the language model. We can also use this model in natural language understanding because most words have more than one linguistic use. Specifically, within a parser, we must allow all possible syntactic derivations of each word while we search for the most likely parse of the whole sentence.

So, for example, if we want to determine the most likely syntactic part of speech or *tag* for each word in a sentence, we can formulate a probabilistic tagging model. Let us assume that we want to know the most likely tag sequence, T, given a particular word sequence, W. Using Bayes' rule we can write the *a posteriori* probability of tag sequence T given word sequence was

$$P(T|W) = \frac{P(T)P(W|T)}{P(W)}$$

where P(T) is the a priori probability of tag sequence T, P(W|T) is the conditional probability of word sequence W occurring given that a sequence of tags T occurred, and P(W) is the unconditioned probability of word sequence W. Then, in principle, we can consider all possible tag sequences, evaluate the a posteriori probability of each, and choose the one that is highest. Since W is the same for all hypothesized tag sequences, we can disregard P(W). Intuitively we have turned the problem around to say, if we knew the tag sequence, what would be the likelihood of each word sequence.

We can rewrite the probability of each sequence as a product of the conditional probabilities of each word or tag given all of the previous tags.

$$P(T|W)\,P(W) = \prod p(t_0) * p(t_i \mid t_{i-1}\, t_{i-2},\ldots) * p(w_i \mid t_i..,w_{i-1}..)$$

Now, we can make the approximation that each tag depends only the immediately preceding tags (say the two preceding tags for a tritag model), and that the word depends only on the tag that it is.

$$P(T|W)\,P(W) = p(t_0) * \prod p(t_i \mid t_{i-1}, t_{i-2}) * p(w_i \mid t_i)$$

That is, once we know the tag that will be used, we gain no further information about the likely word from knowing the previous tags or words. This model is called a Markov model, and the assumption is frequently called the Markov independence assumption.

If we have sufficient training data then we can estimate the tag n-gram sequence probabilities and the probability of each word given a tag (lexical probabilities). We use robust estimation techniques that take care of the cases of unobserved events.

However, in real-world problems, we also are likely to have words that were never observed at all in the training data. The model given above can still be used, simply by defining a generic new word called "unknown-word". The system can then guess at the tag of the unknown word primarily using the tag sequence probabilities.

However, we can frequently guess the tag of a new word from its orthographic spelling, which frequently has different endings for different types of words. For example, a word ending in "-ing" is probably a verb, while a word ending in "-tion" is probably a noun. In addition, a non-initial word that is capitalized is probably a proper noun.

We can incorporate these features of the word into the probability that this particular word will occur given a particular tag using

$$p(w_j \mid t_i) = p(\text{unknown-word} \mid t_i) * p(\text{Capital - feature} \mid t_i) * p(\text{ending} \mid t_i)$$

We estimate the probability of each of 32 endings for each tag based on the training data. While these probabilities are not strictly independent, the approximation is good enough to make a marked difference in classification of unknown words. As the experiment in section 7 shows, the use of the orthographic endings of the words reduces the error rate on the unknown words by a factor of 3.

5.2 Training the models

Simple but powerful models to predict part of speech can be derived using a corpus that has been tagged (or labelled) as to part of speech [Church 1988; de Marken 1990]. Using a tagged corpus to train the model is called "supervised training", since a human has prepared the correct training data. This is in contrast to "unsupervised training" where the process is fully automated.

For example, in unsupervised part of speech tagging, one can use a corpus without annotation for training, a dictionary that lists parts of speech for the most frequently occurring words, and an initial probability assignment, e.g., a uniform probability distribution or probability estimates from a previous, related domain. When sufficient training data is available, then supervised training is preferred since the resulting model is more accurate and the training takes less time.

We conducted supervised training to derive both a *bi-tag* and a *tri-tag model* based on a corpus from the University of Pennsylvania. The UPenn corpus, which was created as part of the TREEBANK project [Santorini 1990] consists of *Wall Street Journal* articles in which each word or punctuation mark has been tagged with one of 47 parts of speech, as shown in the following example:

Terms/NNS were/VBD not/RB disclosed/VBN ./.[1]

A bi-tag model predicts the relative likelihood of a particular tag given the preceding tag, e.g. how likely is the tag NNS on the third word in the above example, given

[1] NNS means plural noun; VBD past tense of a verb; RB, an adverb; and VBN, the past participle of a verb.

that the previous word was tagged. A tri-tag model predicts the relative likelihood of a particular tag given the two preceding tags, e.g. how likely is the tag RB on the third word in the above example, given that the two previous words were tagged NNS and VBD. While the bi-tag model is faster at processing time, the tri-tag model has a lower error rate. We will use the tri-tag models for the purpose of discussion henceforth.

Using the UPenn corpus, we counted for each possible pair of tags, the number of times that the pair was followed by each possible third tag, and then derived from those counts a probabilistic tri-tag model. We also estimated from the training data the conditional probability of each particular word given a known tag (e.g., how likely is the word to be "terms" if the tag is NNS); this is called the "word emit" probability. Both of these probability estimates used *padding* to an arbitrary estimate to avoid setting the probability for unseen tri-tags or unseen word senses to zero.

Given these probabilities, one can then predict the maximum-likelihood tag sequence for a given word sequence. Using the tri-tag probabilities, we computed the probabilities of all possible paths in the tag space through the sentence, selected the path whose overall probability was highest, and then took the tag predictions from that path. We replicated the result [Church 1988] that this process is able to predict the parts of speech with only a 3-4% error rate when the possible parts of speech of the words are known. This is in fact about the rate of discrepancies among human taggers on the TREEBANK project [Marcus, Santorini & Magerman 1990]. The more interesting result, however is the accuracy of part of speech tagging for words that are a priori unknown to the system.

5.3 Quantity of training data

While supervised training is shown here to be very effective, it requires a correctly tagged corpus. We have done some experiments to quantify how much tagged data is really necessary, and to suggest ways to handle new words when using such models.

In these experiments, we demonstrated that the training set can, in fact, be much smaller than might have been expected. One rule of thumb suggests that the training set needs to be large enough to contain 10 instances of each type of tag sequence in order for their probabilities to be estimated with reasonable accuracy. This would imply that a tri-tag model using 47 possible parts of speech would need a bit more than a million words of training. However, we found that much less training data was necessary, as illustrated in Figure 1.

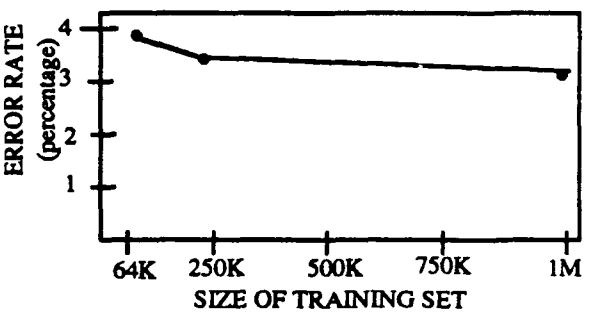


Figure 5-1: Size of Tri-tag Training Sets

In our experiments, the error rate for a supervised tri-tag model increased only from 3.30% to 3.87% when the size of the training set was reduced from 1 million words to 64,000 words. This is probably because most of the possible tri-tag sequences never actually appear.

All that is really necessary, recalling the rule of thumb, is enough training to allow for 10 of each of the tag sequences that do occur. There were 16,170 unique triples in our training set, so the rule of thumb would suggest that 160,000 words would be sufficient training. This would explain why the degradation in performance was slight when the size of the corpus was reduced.

The benefits of probabilistic modeling therefore seem applicable to new tag sets,

subdomains, or languages without needing prohibitively large corpora.

5.4 Unknown words

Sources of open-ended text, such as a newswire, present natural language processing technology with a major challenge: what to do with words the system has never seen before. Current technology depends on handcrafted linguistic and domain knowledge. For instance, the system that performed most successfully in the evaluation of software to extract data from text at the 2nd Message Understanding Conference held at the Naval Ocean Systems Center, June, 1989, would simply halt processing a sentence when a new word was encountered. Determining the part of speech of an unknown word can greatly help the system to know how the word functions in the sentence, for instance, that it is a verb stating an action or state of affairs, that it is a common noun stating a class of persons, places, or things, that it is a proper noun naming a particular person, place, or thing, etc. If it can do that well, then more precise classification and understanding is feasible.

Using the UPenn set of parts of speech, unknown words can be in any of the 22 open-class parts of speech. The tri-tag model can be used to estimate the most probable one. Random choice among the 22 open classes would be expected to show an error rate for new words of 91.5%. The best previously reported error rate was 75% [Kuhn & de Mori 1990].

In our first tests using the tri-tag model we showed an error rate of only 51.6%. However, this model only took into account the context of the word, and no information about the word itself. In many languages, including English, the word endings give strong indicators of the part of speech. Furthermore, capitalization information, when available, can help to indicate whether a word is a proper noun.

We developed a probabilistic model that takes into account features of the word in determining the likelihood of the word given a part of speech. This was used instead of the "word emit" probabilities for known words that the system obtained from training. To develop the model, we first determined the features we thought would distinguish parts of speech. There are four independent[2] categories of features: inflectional endings, derivational endings, hyphenation, and capitalization. Our initial test had three inflectional endings (-ed, -s, -ing), and 32 derivational endings, (including -ion, -al, -ive, -ly). We tested capitalization separately, since some of the data we work with comes only in upper case, obviating the capitalization feature. Capitalization has four values, in our system (+initial +capitalized, -initial +capitalized, etc) in order to take into account the first word of a sentence. The results of the tests were as follows:

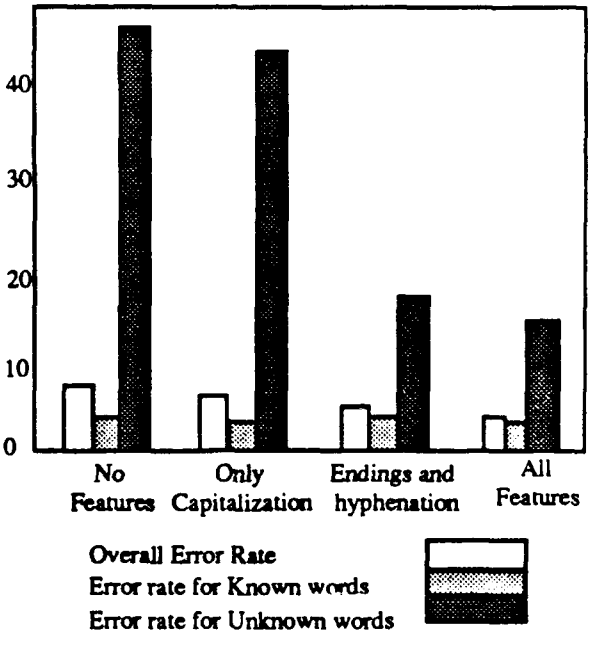


Figure 5-2 Decreasing error rate with use of word features

In sum, adding a probability model of typical endings of words has yielded an

[2] These are not necessarily independent, though we are treating them as such for our tests.

accuracy of 82%. Adding a model of capitalization to the other two models further increased accuracy to 85% for unknown words. The total effect of BBN's model has been a reduction of a factor of five in the error rate of the best previously reported performance for tagging unknown words.

5.5 K-best Tag Sets

An alternative mode of running POST is to return the set of most likely tags, rather than a single tag for each word.

As described earlier, in our first test, the system returned the string of most likely tags for the sentence. This has the advantage of eliminating ambiguity: there is only one tag per word, rather than an average of 2 tags per word for known words and as many as 22 for unknown words. However, even with a rather low error rate of 3.7%, there are cases in which the system returns the wrong tag, which can be fatal for a parsing system.

We addressed this problem by adding the ability of the tagger to return for each word an ordered list of tags, marked by their probability. The following example shows k-best tagging output, with the correct tag for each word marked in bold. Note that the probabilities are in logorithms.

Bailey Controls, based in Wickliffe Ohio, makes computerized industrial controls systems.

Bailey (NP . -1.17) (RB . -1.35) (FW . -2.32) (NN . -2.93) (NPS . -2.95) (JJS . -3.06) (JJ . -3.31) (LS . -3.41) (JJR . -3.70) (NNS . -3.73) (VBG . -3.91)...
Controls (VBZ . -0.19) (NNS . -1 93) (NPS . -3.75) (NP . -4.97)
based (VBN . -0.0001)
in (IN . -.001) (RBV . -707) (NP . -9.002)
Wickliffe (NP . -0.23) (NPS . -1.54)
Ohio (NP . -0.0001)
makes (VBZ . -0.0001)
computerized (VBN . -0.23) (JJ . -1.56)
industrial (JJ . -0.19) (NP . -1.73)
controls (NNS . -0.18) (VBZ . -1.77)
systems (NNS . -0.43) (NPS . -1.56) (NP . -1.95)

Figure 5-3 K-best Tags and Probabilities

In two of the words ("Controls" and "computerized") the first tag is not the correct one. However, in all instances the correct tag is found. Note the first word, "Bailey", is unknown to the system, therefore, all of the open class tags are possible.

In order to reduce the ambiguity further, we tested various ways to limit how many tags were returned based on their probabilities. This is especially useful in cases where one tag is very likely and the others, while possible, are given a low probability, as in the word "in" above. For example, one test used a cut off within some epsilon of the most likely tag. So only tags within 2.0 of the most likely would be included (i.e. if the most likely tag had a probability of -0.19, only tags with a probability of less than -2.19 would be included). This reduced the ambiguity for known words from 1.93 tags per word to 1.23. and for unknown words, from 15.2 to 2.0.

However, the negative side of using cut offs is that the correct tag may be excluded. Note that a cut off of 2.0 would exclude the correct tag for the word "Controls" above. By changing the cut off to 4.0, we are sure to include all the correct tags in this example, but the ambiguity for known words raises to 1.24 and for unknown words to 3.7, for an ambiguity rating of 1.57 overall.

We are continuing experiments to determine the most effective way of limiting the number of tags returned, and hence decreasing ambiguity, while ensuring that the correct tag is in the set.

5.6 Moving to a New Domain

In all of the tests discussed so far, we both trained and tested on sets of articles in the same domain, the Wall Street Journal texts used in the Penn Treebank Project. However, an important measure of the usefulness of the system is how well it performs in other domains. While we would not expect high performance in radically different kinds of

text, such as transcriptions of conversations or technical manuals, we would hope for similar performance on newspaper articles from different sources and on other topics.

We tested this hypothesis using data from the Third Message Understanding Conference (MUC-3). The goal of MUC-3 is to extract data from texts on terrorism in Latin American countries. The texts are mainly newspaper articles, although there are some transcriptions of interviews and speeches. The University of Pennsylvania TREEBANK project tagged four hundred MUC messages (approximately 100,000 words), which we divided into 90% training and 10% testing.

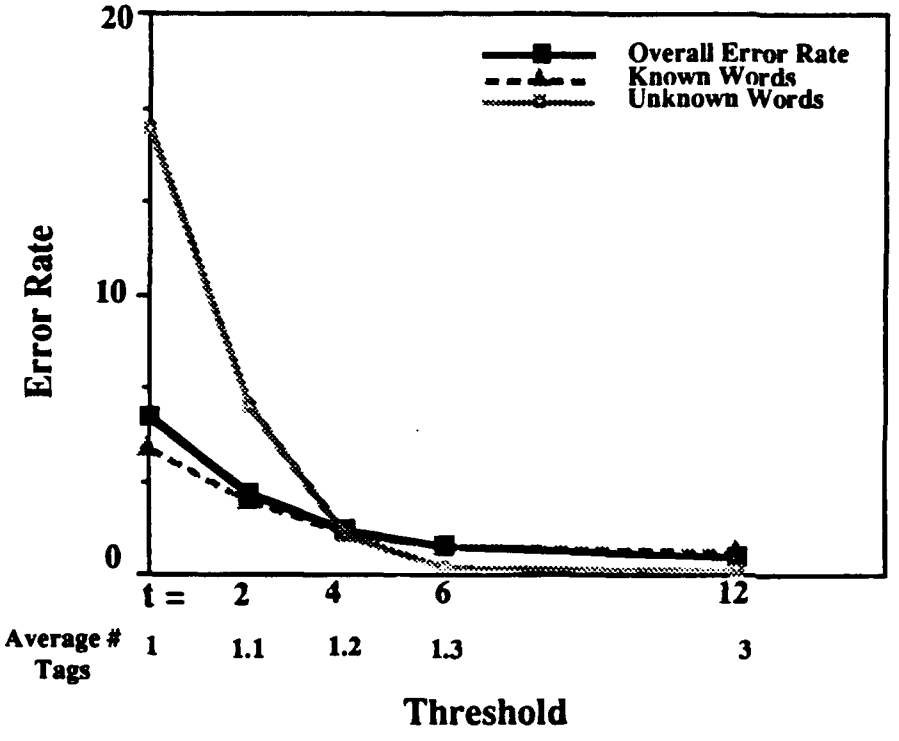


Figure 5-4: **Comparison of thresholds for K- Best**

For our first test, we used the original probability tables trained on the Wall Street Journal articles. We then retrained the probabilities on the MUC messages and ran a second test, with an average improvement of three percentage points in both bi- and tri-tags. The full results are shown below:

BITAGS:	TEST 1	TEST 2
Overall error rate:	8.5	5.6
Number of correct tags:	10340	10667
Number of incorrect tags:	966	639
Error rate for known words:	6.3	4.6
Error rate for unknown words:	25	16
TRITAGS:		
Overall error rate:	8.3	5.7
Number of correct tags:	10358	10651
Number of incorrect tags:	948	655
Error rate for known words:	5.9	4.6
Error rate for unknown words:	26	18

Figure 5-5 **Comparison of original and trained probabilities**

While the results using the new tables are an improvement in these first-best tests, we

saw the best results using K-best mode, which obtained a .7% error rate. We ran several tests using our K-best algorithm with various thresholds. As described in Section 4, the threshold limits how many tags are returned based on their probabilities. While this reduces the ambiguity compared to considering all possibilities, it also increases the error rate. Figure 4 shows this tradeoff from effectively no threshold, on the right hand side of the graph (shown in the figure as a threshold of 12), which has a .7% error rate and an ambiguity of 3, through a cut off of 2, which has a error rate of 2.9, but an ambiguity of nearly zero--i.e. one tag pre word. (Note the far left of the graph is the error rate for a cut off of 0, that is, only considering the first of the k-best tags, which is approximately the same as the bi-tag error rate shown in Figure 3.)

5.7 Using Dictionaries

In all of the results reported here, we are using word/part of speech tables derived from training, rather than on-line dictionaries to determine the possible tags for a given word. The advantage of the tables is that the training provides the probability of a word given a tag, whereas the dictionary makes no distinctions between common and uncommon uses of a word. The disadvantage of this is that uses of a word that did not occur in the training set will be unknown to the system. For example, in the training portion of the WSJ corpus, the word "put" only occurred as verb. However, in our test set, it occurred as a noun in the compound "put option". Since for efficiency reasons, we only consider those tags known to be possible for a word, this will cause an error.

We are currently integrating on-line dictionaries into the system, so that alternative word senses will be considered, while still not opening the set of tags considered for a known word to all open class tags. This will not completely eliminate the problem, since words are often used in novel ways, as in this example from a public radio plea for funds: "You can Mastercard your pledge.". We will be rerunning the experiments reported here to evaluate the effect of using on-line dictionaries.

5.8 Future Directions

In the work reported here, we have evaluated POST in the laboratory, comparing its results against the work of people doing the same task. However, the real test of such a system is how well it functions as a component in a larger system. Can it make a parser work faster and more accurately? Can it help to extract certain kinds of phrases from unrestricted text?

We are currently running these experiments by making POST a part of existing systems. It is being run as a preprocessor to Grishman's Proteus system for the MUC-3 competition [Grishman & Sterling 1989]. Preliminary results showed it sped up Proteus by a factor of two in one-best mode and by a factor of 33% with a threshold of T=2.

POST is also being integrated into a new message processing system at BBN. The results of these experiments will provide us with new directions and ideas both for improving POST and for other ways to integrate probabilistic models into natural language processing systems.

6. SELECTING AMONG INTERPRETATIONS

The performance of today's message processing systems is hindered by the following three complementary problems:

1) Frequently more than one interpretation remains even after all linguistic and domain knowledge has been used in processing an input.

2) Partial interpretation, when no complete interpretation can be found, is minimal.

3) Finding any interpretation if the input includes an unknown word

Our results on problems (1) and (3) above are presented in Sections 8.1 and 8.2. The problem of partial interpretation when no complete interpretation can be found is a focus of the second half of this pilot study; our technical direction and our work thus far on that problem appear in Section 9.

6.1 Context-free Models

Probabilities can also quantify the likelihoods of alternative complete interpretations of a sentence. In these experiments, we used the grammar of the Delphi component from BBN's HARC system [Stallard 1989], which combines syntax and semantics in a unification formalism. We developed a *context-free* model, which estimates the probability of each rule in the grammar independently (in contrast to a context-sensitive model, such as the tri-tag model described above, which bases the probability of a tag on what other tags are in the adjacent context).

In our context-free model, we associate a probability with each rule of the grammar. For each distinct major category (left-hand side) of the grammar, there is a set of context-free rules

$$LHS \leftarrow RHS_1$$
$$LHS \leftarrow RHS_2$$
$$\bullet\bullet\bullet$$
$$LHS \leftarrow RHS_n.$$

For each rule, we estimate the probability of the right-hand side given the left-hand side.

The probability of a syntactic structure S, given the input string W, is then modelled by the product of the probabilities of the rules used in S. ([Chitrao & Grishman 1990] used a similar context-free model.)

6.2 Resolving Ambiguity in Interpretation

Using a context-free probability model, we explored the following issues:

- What method of training the rule probabilities should be employed?
- How much training data is required for reliable estimates?
- How is system performance impacted?
- Do the results suggest refinements in the probability model?

Our intention is to use the Treebank corpus being developed at the University of Pennsylvania as a source of correct structures for training. However, until that material becomes available, we have run initial experiments using small training sets taken from an existing question-answering corpus of sentences about a personnel database. To our surprise, we found that as little as 100 sentences of supervised training (in which a person, using graphical tools, identifies the correct parse) is sufficient to improve the ranking of the interpretations found.

In our tests, the NLP system produces all interpretations satisfying all syntactic and semantic constraints. From that set, the intended interpretation must be chosen. The context-free probability model reduced the error rate on independent test sets by factors of two to four, compared to random selection

from the interpretations satisfying all knowledge-based constraints.

We tested the predictive power of rule probabilities using this model both in unsupervised and in supervised mode. In the former case, the input is all parse trees for the sentences in the training set (even though some parse trees will not be correct in the unsupervised case). In the latter case, supervised under the training data included a specification of the correct parse as hand picked by the grammar's author from among the parse trees produced by the system.

The detailed results from using a training set of 81 sentences appear in the histogram in Figure 6-1.

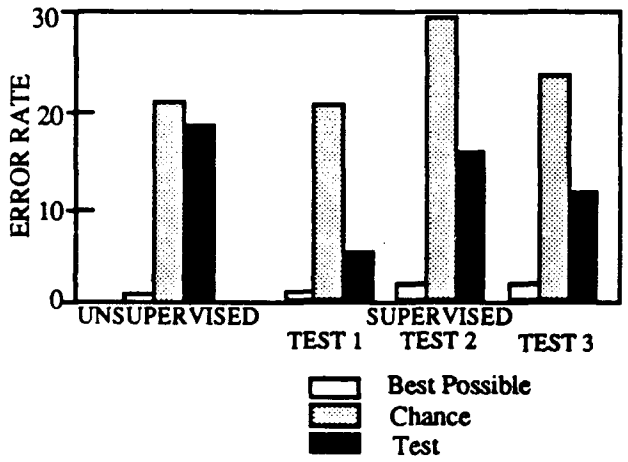


Figure 6-1: **Predictions of Probabilistic Language Model**

The "best possible" error rates for each test indicates the percentage of cases for which none of the interpretations produced by the system was judged correct, so that no selection scheme could achieve a lower error rate than that. The "chance" score gives the error rate that would be expected with random selection from all interpretations produced. The "test" column shows the error rate with the supervised or unsupervised probability model in question. The first supervised test had an 81.4% improvement, and the second a 50.8% improvement, and the third a 56% improvement. These results state how much better than chance the given model did as a percentage of the maximum possible improvement.

We expect that the model's performance can by improved by recording probabilities for other features in addition to just the set of rules involved in producing them. For example, in the grammar used for this test, two different attachments for a prepositional phrase produced trees with the same set of rules, but differing in shape. Thus the simple, context-free model based on the product of rule probabilities could not capture preferences concerning such attachment. By adding to the model probabilities for such additional features, we expect that the power of the probabilistic model to automatically select the correct parse can be substantially increased.

6.3 Experiment in Parsing with Unknown Words

One purpose for probabilistic models is to contribute to handling new words or partially understood sentences. We have done preliminary experiments that show that there is promise in learning lexical syntactic and semantic features from context when probabilistic tools are used to help control the ambiguity.

In our experiments, we used a corpus of sentences each with one word that the system did not know. To create the corpus, we began with a corpus of sentences known to parse from the personnel question-answering domain (our goal, again, is to use the Treebank data from the University of Pennsylvania for such training when it becomes available). We then replaced one word in each sentence with an undefined word.

For example, in the following sentence, assume that the word "contact" is undefined in the system: *Who in Division Four is the contact for MIT*? The word "contract" has both a noun and a verb part of speech; however, the pattern of parts of speech of the words surrounding "contact" causes the tri-tag

model to return a high probability that the word is a noun. Section 7.4 presented experimental performance for this part of speech tagging on such unknown words. Using unification variables for all possible features of a noun, the parser produces multiple parses. Applying the context-free rule probabilities to select the most probable of the resulting parses allows the system to conclude both syntactic and semantic facts about "contact". Syntactically, the system discovers that it is a count noun, with third person singular agreement. Semantically, the system learns (from the use of who) that "contact" is in the semantic class PERSONS.

Furthermore, the partially-specified semantic representation for the sentence as a whole also shows the semantic relation to SCHOOLS, which is expressed here by the *for* phrase. Thus, even a single use of an unknown word in context can supply useful data about its syntactic and semantic features.

Probalistic modelling plays a key role in this process. While context sensitive techniques for inferring lexical features can contribute a great deal, they can still leave substantial ambiguity. As a simple example, suppose the word "list" is undefined in the sentence "List the employees." The tri-tag model predicts both a noun and a verb part of speech in that position. Using an underspecified noun sense combined with the usual definitions for the rest of the words yields no parses. However, an underspecified verb sense yields three parses, differing in the subcategorization frame of the verb "list". For more complex sentences, even with this very limited protocol, the number of parses for the appropriate word sense can reach into the hundreds.

Using the rule probabilities acquired through supervised training (described in the previous section), the likelihood of the ambiguous interpretations resulting from a sentence with an unknown word was computed. Then we tested whether the tree ranked most highly matched the tree previously selected by a person as the correct one. This tree equivalence test was based on the trees' structure and on the rule applied at each node; while an underspecified tree might have some less-specified feature values than the chosen fully-specified tree, it would still be equivalent in the sense above.

Of 160 inputs with an unknown word, in 130 cases the most likely tree matched the correct one, for an error rate of 18.75%, while picking at random would have resulted in an error rate of 63.14%, for an improvement by better than a factor of 3. This suggests that probabilistic modeling can be a powerful tool for controlling the high degree of ambiguity in efforts to automatically acquire lexical data.

We have also begun to explore heuristics for combining lexical data for a single word acquired from a number of partial parses. There are some cases in which the best approach is to unify the two learned sets of lexical features, so that the derived sense becomes the sum of the information learned from the two examples. For instance, the verb subcategorization information learned from one example could be thus combined with agreement information learned from another. On the other hand, there are many cases, including alternative subcategorization frames, where each of the encountered options needs to be included as separate alternatives.

7. PARTIAL PARSING

Traditional NLP techniques have focussed on obtaining complete syntactic analyses of all linguistic input. However, complete analysis of all input is probably an unattainable goal for processing large amounts of text. Grammars are always incomplete, text often contains new lexical items, and there are errors in the production, transmission, or reception of the text. In addition, insistence on complete syntactic analysis is usually unnecessary, since much of the input isn't strictly relevant to the message processing task.

An alternative to traditional parsers is represented in FIDDITCH [Hindle, 1983], MITFP [deMarcken 1990], and CASS [Abeny, 1990]. Instead of requiring complete parses, a forest is frequently produced, each tree in the forest representing a non-overlapping fragment of the input. However, algorithms for finding the semantics of the whole from the disjoint fragments have not previously been developed nor evaluated.

Recent research at BBN has moved away from the goal of complete analysis, to a more realistic one of extracting only the task-appropriate meaning from the input. This involves determining three components of the input:

- the entities (what is being talked about)
- the relations between the entities (what relations are explicit in the input)
- the "total picture" (relations to other entities in discourse, in the domain model, etc.)

The question this research is trying to answer is how well the linguistic expression of the entities and the relational structures among them can be recovered, without determining global syntactic structure by exhaustive analysis.

We are comparing several differing algorithms from various sites to evaluate both the effectiveness of such a strategy in correctly predicting fragments and the effectiveness of syntactic/semantic algorithms for combining fragments. One question our research is trying to answer is how well the linguistic expression of entities and the relational structures among them can be recovered for data base update without determining global syntactic structure and without full information regarding the vocabulary items. A second question is how well an algorithm to learn lexical semantic knowledge from examples will perform.

7.1 Application Context

For message processing, insistence on complete syntactic analysis is usually unnecessary, since much of the input isn't directly relevant to updating a data base, routing a message, or prioritizing it.

An example article from the Third Message Understanding Conference (MUC-3), illustrates how complete analysis is unnecessary; the first sixteen paragraphs relate the results of a summit between the presidents of Peru and Bolivia. Those paragraphs would not add anything to the MUC-3 data base on terrorist acts. However, the final two sentences of the article mention, almost incidentally, that a bomb exploded near the summit, and therefore, do provide data to be added to the terrorism data base.

Even at the sentential level, one is not likely to be able to reliably compute a full semantic interpretation. For instance, in the sentence below from the aforementioned article, only the material in italics actually contributes to the desired, pre-defined data base update:

A BOMB EXPLODED TODAY AT DAWN *IN THE PERUVIAN TOWN OF YUNGUYO,* NEAR THE LAKE, VERY NEAR WHERE THE PRESIDENTIAL SUMMIT WAS TO TAKE PLACE.

Figure 7-1 A Sentence from the MUC-3 Corpus

In a task such as MUC-3 the goal is to identify pre-defined classes of entities, e.g., terrorist events, and dates, and the relationships among them, e.g., the perpetrator of a given terrorist act. Below, we have listed

the first seven of nineteen pre-specified classes of data to be extracted from the messages of MUC-3.

0. MESSAGE ID: identifier
1. TEMPLATE ID: identifier
2. DATE OF INCIDENT: date
3. TYPE OF INCIDENT: set element e.g. KIDNAPPING, ATTEMPTED KIDNAPPING, KIDNAPPING THREAT,...
4. CATEGORY OF INCIDENT: set element, e.g, TERRORIST ACT, STATE-SPONSORED VIOLENCE
5. PERPETRATOR: ID OF INDIV(S): a string
6. PERPETRATOR: ID OF ORG(S): a string

7.2 Finding Core Noun Phrases

In a task such as MUC-3 one fundamental application goal is to identify pre-defined classes of entities, e.g., dates, locations, individuals, and organizations of primary interest in the domain. Normally these entities appear as noun phrases in the text. Therefore, a basic concern is to reliably identify noun phrases that denote entities of interest, even if neither full syntactic nor full semantic analysis is possible.

Two of our experiments have focussed on the identification of core noun phrases, a primary way of expressing entities in text. A core NP is defined syntactically as the maximal simple noun phrase, i.e., the largest one containing no post-modifiers. Here are some examples of core NPs (marked by italics) within their full noun phrases:
a joint venture with the Chinese government to build an automobile-parts assembly plant

a $50.9 million loss from discontinued operations in the third quarter because of the proposed sale

Such complex, full NPs require too many linguistic decisions to be directly processed without detailed syntactic and semantic knowledge about each word, an assumption which need not be true for open-ended text.

We tested two differing algorithms on text from the *Wall Street Journal* (WSJ). Using BBN's part of speech tagger (POST), tagged text was parsed using the full unification grammar of Delphi to find only core NPs, 695 in 100 sentences. Hand-scoring of the results indicated that 85% of the core NPs were identified correctly. Subsequent analysis suggested that half the errors could be removed with only a little additional work, suggesting that over 90% performance is achievable.

In a related test, we explored the bracketings produced by Church's PARTS program [Church, 1988]. We extracted 200 sentences of WSJ text by taking every tenth sentence from a collection of manually corrected parse trees (data from the TREEBANK Project at the University of Pennsylvania). We evaluated the NP bracketings in these 200 sentences by hand, and tried to classify the errors. Of 1226 phrases in the 200 sentences, 131 were errors, for a 10.7% error rate. The errors were classified by hand as follows:

Frequency	Category
10	Two consecutive but unrelated phrases grouped as one
70	Phrase consisted of a single word, which was not an NP
12	Missed phrases (those that should have been bracketed but were not)
4	Ellided head (e.g. part of a conjoined premodifier to an NP)
4	Missed premodifiers
4	Head of phrase was verb form that was missed
27	Other

The 90% success rate in both tests

suggests that identification of core NPs can be achieved using only local information and with minimal knowledge of the words. Next we consider the issue of what semantics should be assigned and how reliably that can be accomplished.

7.3 Semantics of Core Noun Phrases

In trying to extract pre-specified data from open-ended text such as a newswire, it is clear that full semantic interpretation of such texts is not on the horizon. However, our hypothesis is that it need not be for automatic data base update. The type of information to be extracted permits some partial understanding. For semantic processing, minimally, for each noun phrase (NP), one would like to identify the class in the domain model that is the smallest pre-defined class containing the NPs denotation. For each clause, one would like to identify the corresponding event class or state of affairs denoted.

Our pilot experiment focussed on the reliability of identifying the minimal class for each noun phrase.

Assigning a semantic class to a core noun phrase can be handled via some structural rules. Usually the semantic class of the head word is correct for the semantic class not only of the core noun phrase but also of the complete noun phrase it is part of. Additional rules cover exceptions, such as "set of ...". These heuristics correctly predicted the semantic class of the whole noun phrase 99% of the time in the sample of over 1000 noun phrases from WSJ that were correctly predicted by Church's PARTS program.

Furthermore, even some of the NP's whose left boundary was not predicted correctly by PARTS, nevertheless were assigned the correct semantic class. One consequence of this is that the correct semantic class of a complex noun phrase can be predicted even if some of the words in the noun phrase are unknown and even if its full structure is unknown. Thus, fully correct identification of core noun phrase boundaries and noun phrase boundaries may not be necessary to accurately produce data base updates.

7.4 Finding Relations/Combining Fragments

Though finding the entities of interest is fundamental to the task, finding relationships of interest among them is also critical. For instance, in MUC-3 one must identify terrorist events in any of nine Latin American countries, the perpetrators of the event, the victims, if any, the date, the location, any structural damage, and so on.

The experiments reported above were run by mid-summer, 1990. In fall, 1990, a more complete alternative, the MIT Fast Parser (MITFP) [deMarcken, 1990], became available to us. It finds fragments using a stochastic part of speech algorithm and a nearly deterministic parser. It produces fragments averaging 3-4 words in length. An example output is given in figure 7-2.

```
(S (NP (DETERMINER "A") (N "BOMB"))
  (VP (AUX (NP (MONTH "TODAY"))
        (PP (PREP "AT")
          (NP (N "DAWN"))))
    (VP (V "EXPLODED"))))
(PP
 (PP (PREP "IN")
   (NP (NP (DETERMINER "THE")
         (N "PERUVIAN")
         (N "TOWN"))
       (PP (PREP "OF")
         (NP (N "YUNGUYO")))))
 (PUNCT ","))
(PP (PP (PREP "NEAR")
     (NP (DETERMINER "THE")
       (N "LAKE")))
  (PUNCT ","))
(ADJP (DEGREESPEC "VERY")
   (ADJP (ADJ "NEAR")))
(ADV "WHERE")
(NP (DETERMINER "THE")
  (ADJP (ADJ "PRESIDENTIAL"))
  (N "SUMMIT"))
(VP (AUX) (VP (V "WAS")))
(VP (AUX "TO")
  (VP (V "TAKE")
```

```
    (NP (N "PLACE"))))
(PUNCT ".")
```

Figure 7-2 Example Output from MITFP

Certain sequences of fragments appear frequently, as illustrated in the tables below. One frequently occurring pair is an S followed by a PP (prepositional phrase). Since there is more than one way the parser could attach the PP, and syntactic grounds alone for attaching the PP would yield poor performance, semantic preferences applied by a post-process that combines fragments are called for.

Pair		Occurrences
S	PP	104
NP	VP	89
VP	VP	72
S	VP	65
PP	PP	62
PP	NP	58
NP	PP	56
VP	PP	54
PP	VP	48
NP	NP	34

Figure 7-3 Most Frequently Occurring Pairs (In 2500 Pairs)

Triple			Occurrences
NP	PUNCT	NP	53
VP	PUNCT	S	20
S	PUNCT	S	19
NP	PUNCT	S	19
S	PUNCT	NP	17
VP	PUNCT	N	12
NP	PUNCT	PP	10
NP	PUNCT	VP	9

Figure 7-4 Frequently Occurring Fragment Pairs Surrounding Punctuation

In our approach, the first step is to compute a semantic interpretation for each fragment found without assuming that the meaning of each word is known. For instance, as described above, the semantic class for any noun phrase can be computed provided the head noun has semantics in the domain.

Based on the data above, a reasonable approach is an algorithm that moves left-to-right through the set of fragments produced by MITFP, deciding to attach fragments (or not) based on semantic criteria. To avoid requiring a complete, global analysis, a window two constituents wide is used to find patterns of possible relations among phrases. For example, an S followed by a PP invokes an action of finding all points along the "right edge" of the S tree where a PP could attach, applying the fragment combining patterns at each such spot, and ranking the alternatives.

As evident in Table 2, MITFP frequently does not attach punctuation. This is to be expected, since punctuation is used in many ways, and there is no deterministic basis grounds for attaching the constituent following the punctuation to the constituent preceding it. Therefore, if the pair being examined by the combining algorithms ends in punctuation, the algorithm looks at the constituent following it, trying to combine it with the constituent left of the punctuation.

A similar case is when the pair ends in a conjunction. Here the algorithm tries to combine the constituent to the right of the conjunction with that on the left of the conjunction.

Since the norm will be that there are several ways to combine a pair of fragments, we plan to test several alternative heuristics for ranking the alternatives. Probabilistic methods seem particularly powerful and appropriate.

Thus far, we have tested this hypothesis on propositional phrase attachment. The experiment is reported in the next chapter.

8. SEMANTIC ANNOTATION AND SEMANTIC ACQUISITION

During this contract we have begun investigating techniques for semantic annotation of phrases. The major aim of this research is to provide a resource for automatic or semi-automatic training of NLP systems. This can be considered a semantic counterpart to the syntactic annotation of the University of Pennsylvania Treebank Project. Semantic annotation of a large enough domain-relevant corpus is prerequisite to training of NLP systems, both the acquisition of symbolic rules (e.g., knowledge) and probabilities on those rules. This approach assumes at least partial parsing of the type described in Section 9 of this report.

Consider the following example sentence fragment from MUC domain of terrorist reports:

The Tupac Amaru Revolutionary Movement (MRTA) is staging attacks in an attempt to ...

If a NLP system already knows that *staging attacks* takes an agent argument, and that agents of such attacks are (with some high probability) terrorists, this provides evidence that core NP *the Tupac Amaru Revolutionary Movement* is actually a terrorist organization. Thus information about the expected arguments of verbs can be used to form predictions about the actual arguments observed in text.

Semantic annotation also provides a useful means of generalizing the contexts in which a word occurs. Referring back to the previous example, the precise lexical environments of a word like *attack* will vary widely. But if the arguments with which it occurs can be grouped according to semantic class, this can provide more focussed training data on verb contexts.

Such semantic knowledge called *selection restrictions* or *case frames* governs what phrases make sense with a particular verb or noun (what arguments go with a particular verb or noun). Traditionally such semantic knowledge is handcrafted, though some software aids exist to enable greater productivity [Ayuso et al., 1987; Bates, 1989; Grishman et al., 1986; Weischedel, et al., 1989].

Instead of handrafting this semantic knowledge, our goal is to learn that knowledge from examples, using a three step process:

1. Simple manual semantic annotation,
2. Supervised training based on parsed sentences,
3. Estimation of probabilities.

8.1 Simple Manual Semantic Annotation

Given a sample of text, we annotate each noun, verb, and proper noun in the sample with the semantic class corresponding to it in the domain model. For instance, *dawn* would be annotated <time>, *explode* would be <explosion event>, and *Yunguyo* would be <city>. For our experiment, 560 nouns and 170 verbs were defined in this way resulting in xxx semantic classes. We estimate that this semantic annotation proceeded at about 90 words/hour.

8.2 Supervised Training

From the TREEBANK project at the University of Pennsylvania, we used 20,000 words of MUC-3 texts that had been bracketed according to major syntactic category. The bracketed constituents for the sentence used in figure 7-1 appears in figure 8-1.

```
((S
    (NP a bomb)
    (VP exploded
        today
        (PP at
            (NP dawn) )
        (PP in
```

```
(NP the Peruvian town
    (PP of
        (NP yunguyo) ) ) )
,
(PP near
    (NP the lake) )
,
(SBAR ( WHPP very
                near
                (WHADVP where) )
        (S (NP the presidential summit)
            (VP was
            (S (NP*)
                to
                (VP take
                    (NP place) ) ) ) ) ) ) )
.)
```

Figure 8-1 Example of TREEBANK Analysis

From the example one can clearly infer that bombs can explode, or more properly, that *bomb* can be the logical subject of *explode*, that *at dawn* can modify *explode*, etc. Naturally good generalizations based on the instances are more valuable than the instances themselves.

Since we have a hierarchical domain model, and since the manual semantic annotation states the relationship between lexical items and concepts in the domain model, we can use the domain model hierarchy as a given set of categories for generalization. However, the critical issue is selecting the right level of generalization given the set of examples in the supervised training set.

We have chosen a known statistical procedure [Katz, 1987] that selects the minimum level of generalization such that there is sufficient data in the training set to support discrimination of cases of attaching phrases (arguments) to their head. This leads us to the next topic, estimation of probabilities from the supervised training set.

8.3 Estimation of Probabilities

The case relation, or selection restriction, to be learned is of the form X P O, where X is a head word or its semantic class; P is a case, e.g., logical subject, logical object, a preposition, etc.; and O is a head word or its semantic class.

One factor in the probability that O attaches to X with case P is p' (X | P, O), an estimate of the likelihood of X given P and O. We chose to model a second factor $p(d)_1$, the probability of an attachment where d words separate the head word X from the phrase to be attached (intuitively, the notion of attachment distance).

Since a 20,000 word corpus is not much data, we used a generalization algorithm [Katz, 1987] to automatically move up the hierarchical domain model from X to its parent, and from O to its parent.

8.4 The Experiment

By examining the table of triples X P O that were learned, it was clear that meaningful information was induced from the examples. For instance, [<attack> *against* <building>] and [<attack> *against* <residence>] were learned, which correspond to two cases of importance in the MUC domain.

However, we ran a far more meaningful evaluation of what was learned by measuring how effective the learned information would be at predicting 166 prepositional phrase attachments that were not made by the MITFP. For example, in figure 8-1, fragment 2 could be attached syntactically to fragment 1 at three places: modifying *dawn*, modifying *today*, or modifying *explode*.

Closest attachment, a purely syntactic

constraint, worked quite effectively, having a 25% error rate. Using the semantic probabilities alone p'(X | P, O) had poorer performance, a 34% error rate. However, the richer probability model p'(X | P, O) * p(d) outperformed both the purely semantic model and the purely syntactic model (closest attachment), yielding an 18% error rate.

As a consequence, useful semantic information was learned by the training algorithm.

However, the degree of reduction of error rate should not be taken as the final word, for the following reasons:

- 20,000 words of training data is much less than one would want. An additional 70,000 words of training data should soon be available through TREEBANK.
- Since many of the head words in the 20,000 word corpus are not of import in the MUC-3 domain, their semantic type is vague, i.e., <unknown event>, <unknown entity>, etc.

8.5 Related Work

In addition to the work discussed earlier on tools to increase the portability of natural language systems, another recent paper [Hindle and Rooth, 1990] is directly related to our goal of inferring case frame information from examples.

Hindle and Rooth focussed only on prepositional phrase attachment using a probabilistic model, whereas our work applies to all case relations. Their work used an unsupervised training corpus of 13 million words to judge the strength of prepositional affinity to verbs, e.g., how likely it is for *to* to attach to the word *go*, for *from* to attach to the word *leave*, or for *to* to attach to the word *flight*. This lexical affinity is measured independent of the object of the preposition. By contrast, we are exploring induction of semantic relations from supervised training, where very little training may be available. Furthermore, we are looking at triples of head word (or semantic class), syntactic case, and head word (or semantic class).

In Hindle and Rooth's test, they evaluated their probability model in the limited case of verb - noun phrase prepositional phrase. Therefore, no model at all would be at least 50% accurate. In our test, many of the test cases involved three or more possible attachment points fro the prepositional phrase, providing a more realistic test.

An interesting next step would be to combine these two probabilistic models (perhaps via linear weights) in order to get the benefit of domain-specific knowledge, as we have explored, and the benefits of domain-independent knowledge, as Hindle and Rooth have explored.

9. DATA REQUIREMENTS ON TRAINING PROBABILISTIC LANGUAGE MODELS

Our new approach assumes there is data to train the system. That data can be used to estimate the probability of certain events; it can also be used to identify events and infer symbolic rules regarding them. We will see both goals in Section 7 regarding part-of-speech assignment; for, we estimate probabilities regarding sequences of parts of speech given a sequence of words, and for words not seen before, we infer what part of speech it occurs in. In either use of the data, a critical issue is how much data is needed.

In observing events, whether linguistic phenomena or not, the most likely events will be seen relatively early. With more and more data, one will still see the most likely events most often, but the chance of seeing a rare event will also be greater. Typically, the number of errors that an algorithm makes due to unobserved events will drop off rapidly initially. However, at some point, the unobserved, rare events will account for most of the error. From then on, increasing the data set greatly will only reduce the error rate slightly, because one needs more and more data to observe a rare event.

Therefore, how much data one needs is really a question of finding the point at which adding data in training does not significantly reduce the error rate.

Consider first the problem of learning rules from observed data. If we see an event once, then there is some probability that it might be an accident or an exception rather than an example of a rule. However, if we see a particular pattern on 10 independent occasions, then the chance that it was accidental is exponentiated to the 10th power, making it most likely that it is the result of a rule.

We will use the number 10 as a reasonable compromise between too little data, and too much training effort. For estimating a probability, if we have observed an event 10 times within some training set, then most likely, if we knew the true probability, the expected number of times it should have occurred would be between 5 and 15. Therefore, our probability is most likely accurate to within a factor of 2. If we have observed fewer than 10 instances of an event, then the ratio of the expected number and the observed number might be larger, and therefore not considered to be accurate.

For some kinds of rules, we only need to see one instance to know that it is possible. For example, if you see a word once and are told that in this instance it has a particular meaning, then you know that this is one of the possible meanings of the word. Let us assume that we will create a rule for each new phenomenon. Then we must observe at least one instance of each kind of phenomenon in order to have all the necessary rules. The question is, how much training is needed to observe at least one instance of enough rules to give performance judged as acceptable.

If we somehow know the number of phenomena that will be possible, then we can estimate how many events will be necessary to have seen most of the possible ones. Again, we might say that when we have seen an average of 10 of each of the events that we have observed, then we have probably observed most of the common events at least once. Thus, there may be many possible events that we haven't observed at all, but they will be quite unlikely, and may not hurt performance much. To get enough data to observe them might require an order of magnitude more data, with minimal improvement in coverage.

We can, at any point, estimate the probability that the next sentence or message will need a rule that we don't have. There are two simple ways to estimate the likelihood that a new event will not be covered by the existing rules. The simplest way is to measure the coverage of a set of new events. The fraction of covered events is a reasonable estimate of future coverage. A second way to estimate coverage is to count how many of the

rules in the system have occurred exactly once in the training set. If we divide this number by the total number of instances of rules used, then this is a good estimate of the likelihood that a new event would not be covered. (This well-known result follows from the logic that if the sentence or message that supplied that rule were the test message, then that rule would be missing.)

In general, in order to estimate probabilities of a known set of events, we must observe several occurrences of each event that actually occurs within our training set. Usually 10 occurrences of an event is considered sufficient for a good estimate. However, we can often get a **useful** estimate of the probabilities from as few as 2 or 3 occurrences of an event. In our experience, an **average** of 10 occurrences per observed event is usually adequate.

It is important to remember that we do not need explicit training data to estimate the probabilities of events that never occur. That is, given one of several smoothing techniques we can estimate the probabilities of the unobserved events roughly from the data we do have. Given that the unobserved events will not occur frequently in test data, we will not be hurt seriously by having only rough estimates of their probabilities. This rule was verified empirically in our work on estimating syntactic category probabilities, as discussed below and in Section 7.

9.1 Syntactic Category Probabilities:

We have performed experiments in which we estimated the probabilities for each triple of syntactic classes and the probability that each word belongs to each syntactic class, in order to be able to determine the syntactic categories of all the words in a new sentence. There were 47 different categories, which would allow for 47**3 or about 100,000 different triples, requiring about 1,000,000 training tokens. However, when we perform an experiment, we notice that only about 10,000 of the triples actually occur, which means that 100,000 training tokens should be sufficient to estimate the probability of each triple. Of course, we must be careful not to assume that the unobserved triples are impossible. When we plotted the accuracy of the assignment of syntactic categories to words we found that the accuracy was only slightly worse when we used 64,000 training words than when we used 250,000 words, which was the same as when we used 1,000,000 words. Thus, we would say that this verifies that 100,000 words is sufficient for estimating probabilities of triples of syntactic categories and the syntactic category probability for each of the words.

9.2 Semantic Knowledge:

The nature of semantic annotation is to identify for each common noun and proper noun, the frame in the domain model representing the class of persons, places or things identified by this word sense; for each verb, the name of the frame in the domain model representing the class of actions or states of affairs identified by its word sense; for each preposition, the slot in the domain model representing the relation corresponding the word sense used in context.

We need to learn three types of semantic knowledge: 1) the possible semantic categories for each word, 2) the groups or sequences of semantic categories that are meaningful, and 3) what each group of semantic categories means. We describe each of these below.

1) The semantic categories in the system will be closely tied to the different domain model concepts and relations. Therefore, once we have derived the domain model, we need to find all the words that could be used to refer to that domain model entity. This can be done by introspection, or more automatically using a thesaurus. It may also happen that there are words that are commonly used in the domain that do not appear with those uses in a thesaurus. We will deal with this in the paragraph on estimating semantic probabilities.

2) For each type of sequence of semantic

categories, there will be some set of possible sequences. For example, for triples of verb-preposition-noun, we must be able to tell whether the types of each word could be used together. If there are V verb categories, P preposition, and N noun categories, then the number of groups that could occur, in principle, is V*P*N. However, most of these will not be possible. Again, we use the rule of 10. When we have seen enough verb phrases with semantic verb-preposition-noun triples so that, on the average we have observed 10 examples of each sequence of types, then any triples that we have not already observed are probably quite unlikely. While we have not discussed it in the proposal, we can represent the probability of each semantic triple, rather than just listing the legal ones. In this case, we can also assume that those that have not been observed are not impossible, but are just quite improbable. It remains, in these cases, to decide what this unobserved triple might mean. However, this is not the topic under discussion here.

There are also several types of pairs and triples of semantic categories of interest in the domain, we must have enough training data to observe them. For example, in addition to verb-preposition-noun, we would have verb-noun, etc. For each type, some fraction F of the possible sequences are actually plausible. So for example, for the verb-preposition-noun type, we would need V*P*N*F samples of this type. We must sum these plausible sequences over the different types, and multiply by W, the average number of words between semantic groups. In order to give some estimate here let's assume that there are T different types and they each have the same number of plausible types as the V*P*N sequence, then the number of words needed would be 10 V*P*N*F*T*W. Assuming, for argument's sake, that V=20, P=10, N=20, F=0.1, T=5, W=10, then we would need 200,000 words.

Another way to estimate the number of sequences of interest is from the complexity of the templates that must be filled, since at least one semantic expression is needed in order to specify the name of each of the fields in the templates. This is discussed further in the section on deriving rules for mapping from domain expressions to templates.

3) Given that we have observed a semantic group of words resulting in an interesting semantic expression, we also assume that the training will contain the meaning of the expression. The meaning for each type of expression would be expected to be the same on each use, so one supervised instance of each expression would be sufficient.

9.3 Semantic Probabilities

Word Meanings

There are several parts to the semantic probabilities. First, let us assume that we have specified the semantic use of each of the interesting words in the training data. We do not need to estimate directly the probability of each of the possible meanings for each word. Instead, we can more easily estimate the probability of using each word to express a particular meaning. Then, the probability of the word given the meaning can be inverted (using Bayes' rule) to construct the probability of the meaning given the word.

For the words appearing in the supervised training data, we can estimate these probabilities simply. If we have 10 tokens on the average word for each meaning, then we should have enough samples to get reasonable probabilities for many words. Next, we consider the words that we assume can be used to express the meaning, based on intuition or a thesaurus. Since we did not observe them in the training data, we can assume their probability (taken as a group) is roughly equal to the probability of all the words that were used once to express this meaning (again taken as a group). For lack of a better model, we must assume that each of these unobserved words is equally likely. Finally, we can consider other words that are neither in the training nor in our thesaurus. For example, words that are related to the meaning of interest might be used to refer to the meaning. The probability of these metonymic references

could be determined from examples, and related to the domain model, or if necessary, could be predicted from the logical distance (in the domain model) between the concept usually referred to by the word and the meaning of interest.

9.4 Semantic Expressions

Another component in the semantic probability is the probability of each type of semantic expression. This can be estimated directly by counting the relative frequency of each type of expression. The amount of data suggested in the section on semantic knowledge would also be sufficient to estimate semantic expression probabilities for most of the expression types. These probabilities are used together with the lexical semantic probabilities to determine the probability that a particular derivation has a particular meaning.

10. ACTIVITIES FOR MUC-3

An important part of our work in message processing is providing support for the Third Message Understanding Conference (MUC-3). The MUC series is designed to foster both cooperation and evaluation among researchers in message processing as numerous sites apply their systems to the same task. All of the sites work with the same training data and then are scored in a final competition on the same set of test data. This provides an objective measure for evaluating the systems that participate in the project.

MUC-3, whose task is to fill data base templates on terrorist acts from open newspaper style text, extends the challenge of the earlier MUCs in both the breadth and complexity of the task. Also new in this latest MUC is the extent of data that is available for training. This large amount of data is allowing participants to try new approaches involving automatic acquisition of knowledge, rather than handcrafting rules.

We are working both with Beth Sundheim at NOSC to create the development corpus that is available to all groups for training, and with Ralph Grishman of NYU to improve the performance of NYU's Proteus system in the final evaluation.

10.1 Participation at the Organizational Level

The contribution of each individual site involved in MUC-3 is making it possible to have a 1300 message corpus for development. In the early stages of this process, BBN responded both with sample filled templates and careful comments aimed at helping to sharpen the definition of the MUC template fills.

In October, in parallel with the other MUC participants, we provided correct template fill for 100 messages. To facilitate this, we built an Emacs-based tool to ensure conformance to the NOSC specifications and to provide much of the formatting automatically. This approach both saved effort and minimized the potential for formatting errors. We have passed this code along to Beth Sundheim of NOSC in the event she may find it useful for future template filling.

In order to be able to effectively score the performance of systems in a project of this magnitude, it is essential that the definition of the template fills be clear and unambiguous. The original specification of the task included detailed, precise descriptions of the legal fills for each slot in the template. However, once we were involved in the information extraction task, it became clear that because of the complexity and ambiguity of the messages, there were many cases not covered.

Even how many events a message describes is often unclear. According to the guidelines provided by Beth Sundheim "a separate instance is defined as one which has a different physical target (ID or type) or identified location." However, consider the following text:

> The Salvadoran National Police reported this morning that unidentified men set off bombs at the Central American Jose Simeon Canas University, UCA, located in Southwestern San Salvador.
>
> According to the report, the attack took place at 0200. One administrative office, one bus, and one electrical transformer were damaged as a result of the explosion of at least four bombs that went off inside the UCA campus.

Does this article describe one bombing event? (there is only one location given), three? (there are three targets listed) or four (there are four bombs mentioned)?

One can never anticipate all of the kinds of questions that will arise as new articles are processed. In order to keep abreast of the questions and address them fairly, Sundheim has formed a MUC Task Group. BBN is an active participant in this process, with a member of our group on the MUC task group to guide template design.

10.2 Participation in System Development

The second aspect of our participation in MUC is working with NYU to provide support components to improve the performance of their Proteus system. While Proteus was the highest scoring system in the MUC-2 competition, it still needs a great deal of work to be able to handle the greater complexity and range of domain in the MUC-3 task. Two areas in particular that BBN can help in are coverage, especially the ability to handle new words and structures it hasn't seen before, and efficiency, being able to process the large numbers of messages in a reasonable amount of time.

In our support of NYU, we have already sent two components, a part of speech tagger and a domain model, and are exploring other ideas for further means of cooperation.

The part-of-speech tagger (POST) (described in Section 7) uses probabalistic models to assign part of speech to words in open text. POST can help Proteus in two ways: first, having part of speech assigned can greatly speed up the parser by reducing the amount of ambiguity. Speed is important for the same reason it is in speech processing: every speed up means more experiments on the training set can be performed and therefore more new algorithms can be explored.

Second, POST can assign part of speech to unknown words with a high degree of accuracy (85%). This can provide sufficient information for the parser to be able to process the sentence. (In the MUC-2 version of Proteus, it simply stopped processing a sentence when it reached an unknown word.)

Our probabalistic models were trained and tested on Wall Street Journal texts tagged in the Penn Treebank Project. In October we began using the tagger on the MUC data. We ran the system on the first 42 messages and then corrected them to determine error rates. Using the original dictionaries, derived from Wall Street Journal annotations by part of speech, there was a 7.9% overall error rate with 22% of the words unknown. We retrained the dictionaries on the corrected data, and combined the result with the original dictionaries, adding over 1000 words, including many common to the MUC domain such as "terrorist", "Kidnapping", "bombing". In a second test using these new dictionaries only 16% of the words were unknown and the overall error rate was reduced to 6.2%. (Additional training data up to 100,000 words should reduce the overall error rate to 3-4%.)

At the end of October BBN POST was documented and sent to Ralph Grishman of New York University along with 42 tagged MUC articles.

It is being run as a preprocessor to Grishman's Proteus system for the MUC-3 competition [Grishman & Sterling 1989]. Preliminary results showed it sped up Proteus by a factor of two in one-best mode and by a factor of 33% with a threshold of T=2. In this second case, where a 33% speedup was measured, the performance of the system was otherwise largely unaffected. Recall decreased by 1%; precision increased by 1%. It is also being integrated into a new message processing system at BBN.

In addition to part of speech tagging, another area of expertise for BBN that can be useful to NYU is building hierarchical knowledge representations for natural language processing systems [Weischedel, 1989]. In our NLP systems the domain model is a link between the lexicon and semantics. The relations expressed both by the hierarchy and in the roles between concepts provide axiomatic information about the domain for expressing selectional restrictions and understanding metonymy.

We finalized an initial version of a domain model for the MUC task. We sent an abbreviated version to NYU to determine whether this would be helpful to their work, and are awaiting their comments.

For the second phase of MUC, BBN will enter its own message processing system, thereby evaluating many of the techniques developed under this contract.

11. CONCLUSIONS

11.1 Concrete Results

Our most important results are summarized as follows:

1. **We obtained a reduction in error rate in selecting the correct interpretation of a sentence by a factor of two compared to no model.** In a typical natural language processing system, a combination of syntactic and semantic contraints are employed to find an interpretation for an input. When these yield more than one interpretation, a message processing system must select one. We used a context-free probability model on supervised training of only 80 sentences to select among the interpretations produced by syntactic-semantic processing. The error rate in selecting the interpretation was only half that of the same system with no probability model.

2. **Much less training data than theoretically required proved adequate.** Using supervised training with a tri-tag probabilistic model, we achieved a 3-5% error rate in picking the correct part of speech on a test set including both known and unknown words. As little as 64,000 words of supervised training data was used. Combinatorial analysis suggests that 1,000,000 words of training data is required. With 1,000,000 words of supervised training, less than a 1% difference in error rate resulted.

3. **We achieved a five-fold reduction in error rate in predicting the part of speech of unknown words.** In processing unknown words, the best error rate on predicted part of speech as reported in the literature is only 75%. Using a tri-gram model, we found an error rate of 50%. Adding an estimate of the probability of a word ending, given the part of speech, reduced the error rate to 18%. Adding a probability estimate factoring in the likelihood of capitalization, given a part of speech, reduced the error rate for unknown words to 15%.

4. ***We demonstrated that probability* models can improve the performance of knowledge-based syntactic and semantic processing.** It is well known that a unification parser can process an unknown word by collecting the assumptions it makes while trying to find an interpretation for a sentence. Adding a context-free probability model improved the unification predictions of syntactic and semantic properties of an unknown word, reducing the error rate by a factor of two compared to no model.

5. **A simple classification algorithm proved quite effective in detecting relevant versus irrelevant articles.** One set of experiments was classification of articles into specific subcategories. We trained a simple classification algorithm to generate a boolean classification tree for each relevant subcategory, that is, a tree which says whether the article is or is not in the category. Given an article, the different classification trees can be applied to it to determine which relevant categories the article is related to. In the following results, "recalled" is the probability that a message in the class would be classified correctly, and "filtered" is the probability that a message not in the class would be classified correctly.

Category	Recalled	Filtered
BOMBING	100% recalled,	83% filtered
MURDER	87% recalled,	53% filtered
KIDNAP	76% recalled,	93% filtered
ARSON	97% recalled,	97% filtered

11.2 Future directions

Three future directions are called for. First, a battery of experiments evaluating the

effectiveness of these algorithms in a concrete data extraction domain is called for. This will not only measure what these techniques can do but will also suggest alternative ways of combining the probabilistic algorithms for syntactic, semantic, discourse, and template processing.

Second, our one experiment with learning semantic information from examples was quite promising. Learning algorithms employing probabilistic models offer great potential to reduce the effort in acquiring knowledge for a new application domain.

Third, a study to plot how algorithms performance improves with more training data and/or how it degrades with less data would be extremely valuable. Our pilot study has already suggested that much less data may be required than previously envisioned.

11.3 Summary

Two traditional approaches to applying natural language processing techniques are complete syntactic analysis and concept-based analysis. In our approach to open-ended text processing, there are three steps:

1. Probabilistically based syntactic analysis produces a forest of non-overlapping fragments, if no single tree can be found.

2. A semantic interpreter assigns semantic representations to the trees of the forest.

3. Fragments are combined using a probability model reflecting both syntactic and semantic preferences.

One of the most innovative aspects of our approach is the automatic induction of semantic knowledge from annotated examples. The use of probabilistic models offers the induction procedure a decision criterion for making generalizations from the corpus of examples.

The partial parsing approach offers an alternative. By finding fragments based only on syntactic knowledge, and by starting a new fragment when a constituent cannot be deterministically attached, one has some partial analysis of the whole input. How to compute semantic analysis for any constituent is well understood in any compositional semantics. An algorithm that combines the semantically interpreted fragments seems to gain the power of semantically guided analysis without sacrificing syntactic analysis. Fragments that cannot be combined can still be employed with discourse processing and script-based expectations to identify the entities and relations among them for data base update.

Our pilot experiments indicate that the approach to text processing and the induction algorithm are both feasible and promising.

REFERENCES

Ayuso, D.M., Bobrow R., MacLaughlin, D., Meteer, M., Ramshaw, L., Schwartz, R. and Weischedel, R. Toward Understanding Text with a Very Large Vocabulary. In *Proceedings of the Speech and Natural Language Workshop*, Morgan-Kaufmann Publishers, Inc. June, 1990.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Trees*. Wadsworth, 1984.

California Statistical Software, Inc. An *Introduction to CART Methodology*. 1985.

Chitrao, M. and Grishman, R. Statistical Parsing of Messages. In *Proceedings of the Speech and Natural Language Workshop*, Morgan-Kaufmann Publishers, Inc. June, 1990.

Chodorow, M.S., Ravin, Y., and Sachar, H.E. A Tool for Investigating the Synonymy Relation in a Sense Disambiguated Thesaurus. *Proceedings of the Second Conference on Applied Natural Language Processing*, Association for Computational Linguistics, 1988, 144-152.

Church, K. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136-143. ACL, 1988.

Church, K., Gase, W.A., Hanks, P., and Hindle, D. Parsing, Word Associations and Typical Predicate-Argument Relations. *Proceedings of the Speech and Natural Language Workshop*, pages 75-81. Oct. 1989.

Crowther, W. A Common Facts Data Base. In *Speech and Natural Language*, pages 89-93. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1989.

DeJong, G.F. Skimming Stories in Real Time: *An Experiment in Integrated Understanding*. Yale University, Research Report No. 158, May 1979.

de Marcken, C.G. Parsing the LOB Corpus. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 243-251. 1990.

Grishman, R. Analyzing Telegraphic Messages. In *Proceedings of the Speech and Natural Language Workshop*, Morgan-Kaufmann Publishers, Inc., pages 204-208, February, 1989.

Hayes, P., Intelligent High-Volume Text Processing Using Shallow, Domain-Specific Techniques *Spring Symposium Series*, American Association of Artificial Intelligence, March 27-29, 1990.

Hindle, D., and Rooth, M. Structural Ambiguity and Lexical Relations. *Proceedings of the Speech and Natural Language Workshop*, Morgan Kaufman Publishers, Inc., 1990, 257-262.

Hirschman, L., Palmer, M., Dowding, J., Dahl, D., Linebarger, M., Passonneau, R., Lang, F., Ball, C., and Weir, C. The PUNDIT Natural Language Processing System. In *Proc. of the Conference on Artificial Intelligence Systems in Government*, Washington, D.C., March 1989.

Isabelle, P., Machine translation at the TAUM group. Paper presented at The ISSCO *Tutorial on Machine Translation, The state of the art*. 1984.

Jacobs, P.S., Text Power and Intelligent Systems *AAAI Spring Symposium Series*, March 27-29, 1990.

Kuhn, R., and De Mori, R., A cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, pages 570-583. 1990.

Marcus, Santorini & Magerman 1990, "First

Steps Towards an Annotated Database of American English" *Readings for Tagging Linguistic Information in a Text Corpus* Langendoen & Marcus, tutorial for the 28 Annual Meeting of the Association for Computational Linguistics.

Neff, M.S., and Boguraev, B.K. Dictionaries, Dictionary Grammars, and Dictionary Parsing. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics* 1989, 91-101.

Santorini, Beatrice. *Annotation Manual for the Penn Treebank Project.* Technical Report. CIS Department. University of Pennsylvania. May 1990.

Stallard, D. Unification-Based Semantic Interpretation in the BBN Spoken Language System. *Proceedings of the Speech and Natural Language Workshop*, pages 39-46. Oct. 1989.

Wilks, Y.. A Tractable Machine Dictionary as a Resource for Computational Semantics. *NAIC 1987 Natural Language Planning Workshop* , pages 97-123. 1987.

APPENDIX A: EXAMPLE PLUM OUTPUT

A.1 Input message paragraph

THE ARCE BATTALION COMMAND HAS REPORTED THAT ABOUT 50
PEASANTS OF VARIOUS AGES HAVE BEEN KIDNAPPED BY TERRORISTS OF THE
FARABUNDO MARTI NATIONAL LIBERATION FRONT [FMLN] IN SAN MIGUEL
DEPARTMENT. ACCORDING TO THAT GARRISON, THE MASS KIDNAPPING TOOK
PLACE ON 30 DECEMBER IN SAN LUIS DE LA REINA. THE SOURCE ADDED THAT
THE TERRORISTS FORCED THE INDIVIDUALS, WHO WERE TAKEN TO AN
UNKNOWN
LOCATION, OUT OF THEIR RESIDENCES, PRESUMABLY TO INCORPORATE THEM
AGAINST THEIR WILL INTO CLANDESTINE GROUPS.

A.2 MITFP output

```
("THE ARCE BATTALION COMMAND HAS REPORTED THAT ABOUT 50 PEASANTS
OF VARIOUS AGES HAVE BEEN BY TERRORISTS OF THE FARABUNDO MARTI
NATIONAL LIBERATION FRONT KIDNAPPED"
 (S
  (NP (3RD)
     (SINGULAR)
     (COMMON-N)
     :CASE
     (NOT-TRACE)
     (DETERMINER (SINGULAR)
           (NONNULLNBAR)
           "THE")
     (ADJP (ADJ "ARCE"))
     (N (SINGULAR) (COMMON-N) "BATTALION")
     (N (SINGULAR) (COMMON-N) "COMMAND"))
  (VP (AUX (V :TENSE "HAS"))
     (VP (V (PAST) "REPORTED")
       (S (COMP "THAT")
         (S
          (NP (3RD)
             (PLURAL)
             (COMMON-N)
             :CASE
             (NOT-TRACE)
             (NP (3RD)
               (PLURAL)
               (COMMON-N)
               :CASE
               (NOT-TRACE)
               (DETERMINER (PLURAL)
```

```
            (NONNULLNBAR)
            (ADV (POSITIVE)
               "ABOUT")
            (DETERMINER (PLURAL)
                  (NONNULLNBAR)
                  (NUM (PLURAL) "50")))
      (N (PLURAL) (COMMON-N) "PEASANTS"))
    (PP (PREP "OF")
      (NP (3RD)
        (PLURAL)
        (COMMON-N)
        :CASE
        (NOT-TRACE)
        (ADJP (ADJ "VARIOUS"))
        (N (PLURAL) (COMMON-N) "AGES"))))
(VP
  (AUX (V :TENSE "HAVE")
    (V :TENSE "BEEN")
    (PP (PREP "BY")
      (NP (3RD)
        (PLURAL)
        (COMMON-N)
        :CASE
        (NOT-TRACE)
        (NP (3RD)
          (PLURAL)
          (COMMON-N)
          :CASE
          (NOT-TRACE)
          (N (PLURAL)
            (COMMON-N)
            "TERRORISTS"))
        (PP (PREP "OF")
          (NP (3RD)
            (SINGULAR)
            (COMMON-N)
            :CASE
            (NOT-TRACE)
            (DETERMINER (SINGULAR)
                  (NONNULLNBAR)
                  "THE")
            (N (SINGULAR)
              (PROPER-N)
              "FARABUNDO"
              "MARTI"
              "NATIONAL"
              "LIBERATION"
              "FRONT"))))))
  (VP (V (PASSIVE) "KIDNAPPED")
    (NP (3RD)
      (SINGULAR)
```

```
                (COMMON-N)
                :CASE
                (TRACE)))))))))
("[" (PUNCT (LEFT-BRACKET) "["))
("FMLN"
 (NP (3RD)
   (SINGULAR)
   (COMMON-N)
   :CASE
   (NOT-TRACE)
   (N (SINGULAR) (PROPER-N) "FMLN")))
("]" (PUNCT (RIGHT-BRACKET) "]"))
("IN SAN MIGUEL DEPARTMENT"
 (PP (PREP "IN")
   (NP (3RD)
     (SINGULAR)
     (COMMON-N)
     :CASE
     (NOT-TRACE)
     (N (SINGULAR)
       (PROPER-N)
       "SAN"
       "MIGUEL")
     (N (SINGULAR) (COMMON-N) "DEPARTMENT"))))
("." (PUNCT (PERIOD) "."))
("THE MASS KIDNAPPING ACCORDING TO THAT GARRISON , ON 30 DECEMBER
TOOK PLACE"
 (S
  (NP (3RD)
    (SINGULAR)
    (COMMON-N)
    :CASE
    (NOT-TRACE)
    (DETERMINER (SINGULAR)
          (NONNULLNBAR)
          "THE")
    (N (SINGULAR) (COMMON-N) "MASS")
    (N (SINGULAR) (COMMON-N) "KIDNAPPING"))
  (VP
   (AUX
    (PP
     (PP (PREP "ACCORDING TO")
       (NP (3RD)
         (SINGULAR)
         (COMMON-N)
         :CASE
         (NOT-TRACE)
         (DETERMINER (SINGULAR)
               (NULLNBAR)
               "THAT")
         (N (SINGULAR) (COMMON-N) "GARRISON")))
```

```
      (PUNCT (COMMA) ","))
     (PP (PREP "ON")
       (NP (3RD)
          (SINGULAR)
          (DATE-NP)
          :CASE
          (NOT-TRACE)
          (NUM (PLURAL) "30")
          (MONTH "DECEMBER"))))
   (VP (V (PAST) "TOOK")
     (NP (3RD)
        (SINGULAR)
        (COMMON-N)
        :CASE
        (NOT-TRACE)
        (N (SINGULAR) (COMMON-N) "PLACE"))))))
("IN SAN LUIS DE LA REINA"
 (PP (PREP "IN")
   (NP (3RD)
     (SINGULAR)
     (COMMON-N)
     :CASE
     (NOT-TRACE)
     (N (SINGULAR)
       (PROPER-N)
       "SAN"
       "LUIS"
       "DE"
       "LA"
       "REINA"))))
("." (PUNCT (PERIOD) "."))
("THE SOURCE ADDED THAT THE TERRORISTS FORCED THE INDIVIDUALS , WHO
WERE TAKEN TO AN UNKNOWN LOCATION"
 (S
  (NP (3RD)
    (SINGULAR)
    (COMMON-N)
    :CASE
    (NOT-TRACE)
    (DETERMINER (SINGULAR)
          (NONNULLNBAR)
          "THE")
    (N (SINGULAR) (COMMON-N) "SOURCE"))
  (VP (AUX)
    (VP (V (PAST) "ADDED")
      (S (COMP "THAT")
       (S
        (NP (3RD)
          (PLURAL)
          (COMMON-N)
          :CASE
```

```
    (NOT-TRACE)
    (DETERMINER (SINGULAR) (NONNULLNBAR) "THE")
    (N (PLURAL) (COMMON-N) "TERRORISTS"))
  (VP (AUX)
    (VP (V (PAST) "FORCED")
      (NP (3RD)
        (PLURAL)
        (COMMON-N)
        :CASE
        (NOT-TRACE)
        (NP (3RD)
          (PLURAL)
          (COMMON-N)
          :CASE
          (NOT-TRACE)
          (DETERMINER (SINGULAR)
                (NONNULLNBAR)
                "THE")
          (N (PLURAL)
            (COMMON-N)
            "INDIVIDUALS"))
        (S (COMP (PUNCT (COMMA) ",")
            (COMP "WHO"))
          (S
           (NP (3RD)
             (SINGULAR)
             (COMMON-N)
             :CASE
             (TRACE))
           (VP (AUX (V :TENSE "WERE"))
             (VP (V (PASSIVE)
                  "TAKEN")
               (NP (3RD)
                 (SINGULAR)
                 (COMMON-N)
                 :CASE
                 (TRACE))
               (PREP "TO")
               (NP (3RD)
                 (SINGULAR)
                 (COMMON-N)
                 :CASE
                 (NOT-TRACE)
                 (DETERMINER (SINGULAR)
                       (NONNULLNBAR)
                       "AN")
                 (ADJP (ADJ "UNKNOWN"))
                 (N (SINGULAR)
                   (COMMON-N)
                   "LOCATION")))))))))))))))
("," (PUNCT (COMMA) ","))
```

```
("OUT" (PARTICLE "OUT"))
("OF THEIR RESIDENCES ,"
 (PP
  (PP (PREP "OF")
    (NP (3RD)
      (PLURAL)
      (COMMON-N)
      :CASE
      (NOT-TRACE)
      (DETERMINER (SINGULAR)
            (NONNULLNBAR)
            "THEIR")
      (N (PLURAL) (COMMON-N) "RESIDENCES")))
  (PUNCT (COMMA) ",")))
("PRESUMABLY TO AGAINST THEIR WILL INCORPORATE THEM"
 (VP
  (AUX (ADV (POSITIVE) "PRESUMABLY")
     (TO "TO")
     (PP (PREP "AGAINST")
       (NP (3RD)
         (SINGULAR)
         (COMMON-N)
         :CASE
         (NOT-TRACE)
         (DETERMINER (SINGULAR)
               (NONNULLNBAR)
               "THEIR")
         (N (SINGULAR) (COMMON-N) "WILL"))))
  (VP (V :TENSE "INCORPORATE")
    (NP (3RD)
      (SINGULAR)
      (COMMON-N)
      :CASE
      (NOT-TRACE)
      (PRO-DET-SPEC (3RD)
            (PLURAL)
            (OBJ)
            "THEM")))))
("INTO CLANDESTINE GROUPS"
 (PP (PREP "INTO")
   (NP (3RD)
     (PLURAL)
     (COMMON-N)
     :CASE
     (NOT-TRACE)
     (ADJP (ADJ "CLANDESTINE"))
     (N (PLURAL) (COMMON-N) "GROUPS"))))
("." (PUNCT (PERIOD) "."))
```

A.3 Semantic representation

```
> (batch-show-semantics msg-0001 *standard-output*)

;; SEMANTICS for #<Msg: DEV-MUC3-0001>

(S
 "THE ARCE BATTALION COMMAND HAS REPORTED THAT ABOUT 50 PEASANTS
OF VARIOUS AGES HAVE BEEN BY TERRORISTS OF THE FARABUNDO MARTI
NATIONAL LIBERATION FRONT KIDNAPPED"
 (?38
  ((KNOWN-ENTITY ?4
          PEOPLE
          (SOCIAL-ROLE-OF ?4 MILITARY)
          (DESCRIPTION-OF ?4
                  "THE ARCE BATTALION COMMAND")
          (DET ?4 "THE"))
   (KNOWN-EVENT ?38
          COMMUNICATION
          (AGENT-OF ?38 ?4)
          (PERFECT-TENSE ?38 "HAS")
          (PRESENT-TENSE ?38)
          (OBJECT-OF ?38 ?35)
          (PAST-TENSE ?38))
   (KNOWN-EVENT ?35
          KIDNAPPING
          (OBJECT-OF ?35 ?13)
          (AGENT-OF ?35 ?29)
          (PERFECT-TENSE ?35 "HAVE")
          (PRESENT-TENSE ?35)
          (PASSIVE ?35))
   (KNOWN-ENTITY ?13
          PERSON
          (PP-MODIFIER ?13 ?12 "OF")
          (SOCIAL-ROLE-OF ?13 CIVILIAN)
          (NUMBER-OF ?13 50)
          (DESCRIPTION-OF ?13 "ABOUT 50 PEASANTS"))
   (KNOWN-SOA ?12
         STATE-OF-AFFAIRS
         (NUMBER-OF ?12 PLURAL)
         (DESCRIPTION-OF ?12 "VARIOUS AGES"))
   (KNOWN-ENTITY ?23
          ORGANIZATION
          (NAME-OF ?23 "FMLN")
          (NAME-OF ?23
                "FARABUNDO MARTI NATIONAL LIBERATION FRONT")
          (SOCIAL-ROLE-OF ?23 TERRORISM)
          (DESCRIPTION-OF ?23
                  "THE FARABUNDO MARTI NATIONAL LIBERATION FRONT")
```

```
            (DET ?23 "THE"))
    (KNOWN-ENTITY ?29
            PERSON
            (PP-MODIFIER ?29 ?23 "OF")
            (SOCIAL-ROLE-OF ?29 TERRORISM)
            (NUMBER-OF ?29 PLURAL)
            (DESCRIPTION-OF ?29 "TERRORISTS")))
  NIL))

(PUNCT "[" NIL)

(NP "FMLN"
  (?40
   ((KNOWN-ENTITY ?40
            ORGANIZATION
            (NAME-OF ?40 "FMLN")
            (NAME-OF ?40
                 "FARABUNDO MARTI NATIONAL LIBERATION FRONT")
            (SOCIAL-ROLE-OF ?40 TERRORISM)
            (DESCRIPTION-OF ?40 "FMLN")))
    NIL))

(PUNCT "]" NIL)

(PP "IN SAN MIGUEL DEPARTMENT"
  (?46
   ((KNOWN-ENTITY ?46
            DEPARTMENT
            (NAME-OF ?46 "SAN MIGUEL")
            (LOCATION-COUNTRY-OF ?46 ?42))
     (KNOWN-ENTITY ?42
            COUNTRY
            (NAME-OF ?42 "EL SALVADOR")))
    ((PREP NIL "IN"))))

(PUNCT "." NIL)

(S "THE MASS KIDNAPPING ACCORDING TO THAT GARRISON , ON 30 DECEMBER
TOOK PLACE"
  (?66
   ((KNOWN-EVENT ?50
          KIDNAPPING
          (DESCRIPTION-OF ?50
                 "THE MASS KIDNAPPING")
          (DET ?50 "THE"))
    (KNOWN-EVENT ?66
            MOVEMENT
            (UNKNOWN-ROLE ?66 ?50)
            (DATE-OF ?66 ?60)
            (PP-MODIFIER ?66 ?55 "ACCORDING TO")
            (OBJECT-OF ?66 ?63)
```

```
          (PAST-TENSE ?66))
    (KNOWN-ENTITY ?63
          LOCATION
          (DESCRIPTION-OF ?63 "PLACE"))
    (KNOWN-ENTITY ?60
          DATE
          (MONTH-OF ?60 12)
          (DAY-OF ?60 30))
    (KNOWN-ENTITY ?55
          PEOPLE
          (SOCIAL-ROLE-OF ?55 MILITARY)
          (DESCRIPTION-OF ?55 "THAT GARRISON")
          (DET ?55 "THAT")))
  NIL))

(PP "IN SAN LUIS DE LA REINA"
  (?71
   ((KNOWN-ENTITY ?71
          TOWN
          (NAME-OF ?71 "SAN LUIS DE LA REINA")
          (LOCATION-COUNTRY-OF ?71 ?68)
          (DESCRIPTION-OF ?71
                "SAN LUIS DE LA REINA"))
    (KNOWN-ENTITY ?68
          COUNTRY
          (NAME-OF ?68 "EL SALVADOR")))
   ((PREP NIL "IN"))))

(PUNCT "." NIL)

(S
 "THE SOURCE ADDED THAT THE TERRORISTS FORCED THE INDIVIDUALS , WHO
WERE TAKEN TO AN UNKNOWN LOCATION"
 (?104
  ((KNOWN-ENTITY ?74
          PEOPLE
          (DESCRIPTION-OF ?74 "THE SOURCE")
          (DET ?74 "THE"))
   (KNOWN-EVENT ?104
          COMMUNICATION
          (UNKNOWN-ROLE ?104 ?74)
          (OBJECT-OF ?104 ?101)
          (PAST-TENSE ?104))
   (UNKNOWN-EVENT ?101
          UNKNOWN-SITUATION
          (UNKNOWN-ROLE ?101 ?79)
          (OBJECT-OF ?101 ?97))
   (KNOWN-ENTITY ?79
          PERSON
          (SOCIAL-ROLE-OF ?79 TERRORISM)
          (NUMBER-OF ?79 PLURAL)
```

```
            (DESCRIPTION-OF ?79 "THE TERRORISTS")
            (DET ?79 "THE"))
    (KNOWN-ENTITY ?97
            PERSON
            (NUMBER-OF ?97 PLURAL)
            (DESCRIPTION-OF ?97 "THE INDIVIDUALS")
            (DET ?97 "THE")))
  NIL))

(PUNCT "," NIL)

(PARTICLE "OUT" NIL)

(PP "OF THEIR RESIDENCES ,"
  (?110
   ((KNOWN-ENTITY ?110
            RESIDENCE
            (NUMBER-OF ?110 PLURAL)
            (DESCRIPTION-OF ?110 "THEIR RESIDENCES")
            (BELONGS-TO ?110 ?105))
    (REF-ENTITY ?105
            ANYTYPE
            (PRONOUN ?105 "THEM")
            (NUMBER-OF ?105 PLURAL)))
   ((PREP NIL "OF"))))

(VP "PRESUMABLY TO AGAINST THEIR WILL INCORPORATE THEM"
  (?122
   ((REF-ENTITY ?120
            ANYTYPE
            (PRONOUN ?120 "THEM")
            (NUMBER-OF ?120 PLURAL))
    (UNKNOWN-EVENT ?122
             UNKNOWN-SITUATION
             (PP-MODIFIER ?122 ?115 "AGAINST")
             (ADV-MODIFIER ?122 "PRESUMABLY")
             (OBJECT-OF ?122 ?120))
    (REF-ENTITY ?111
            ANYTYPE
            (PRONOUN ?111 "THEM")
            (NUMBER-OF ?111 PLURAL))
    (UNKNOWN-ENTITY ?115
             UNKNOWN-THING
             (DESCRIPTION-OF ?115 "THEIR WILL")
             (BELONGS-TO ?115 ?111)))
   NIL))

(PP "INTO CLANDESTINE GROUPS"
  (?125
   ((KNOWN-ENTITY ?125
            PEOPLE
```

```
            (NUMBER-OF ?125 PLURAL)
            (DESCRIPTION-OF ?125
                    "CLANDESTINE GROUPS")))
    ((PREP NIL "INTO"))))

(PUNCT "." NIL)
```

A.4 Event structure

```
*** Event:
(KIDNAPPING (?50 ?35)
        (TI-PERP-OF (?29 1) (?23 1))
        (EVENT-TIME-OF (?60 1))
        (OBJECT-OF ?13)
        (EVENT-LOCATION-OF (?63 1))
        (TI-INSTRUMENT-OF))
 where:
 Variables (?50 ?60 ?63) in fragment: <S: (THE MASS KIDNAPPING ACCORDING TO
THAT GARRISON , ON 30 DECEMBER TOOK PLACE)>
(?66
 ((KNOWN-EVENT ?50
          KIDNAPPING
          (DESCRIPTION-OF ?50 "THE MASS KIDNAPPING")
          (DET ?50 "THE"))
  (KNOWN-EVENT ?66
          MOVEMENT
          (UNKNOWN-ROLE ?66 ?50)
          (DATE-OF ?66 ?60)
          (PP-MODIFIER ?66 ?55 "ACCORDING TO")
          (OBJECT-OF ?66 ?63)
          (PAST-TENSE ?66))
  (KNOWN-ENTITY ?63
          LOCATION
          (DESCRIPTION-OF ?63 "PLACE"))
  (KNOWN-ENTITY ?60
          DATE
          (MONTH-OF ?60 12)
          (DAY-OF ?60 30))
  (KNOWN-ENTITY ?55
          PEOPLE
          (SOCIAL-ROLE-OF ?55 MILITARY)
          (DESCRIPTION-OF ?55 "THAT GARRISON")
          (DET ?55 "THAT")))
 NIL)
 Variables (?35 ?29 ?23 ?13) in fragment: <S: (THE ARCE BATTALION COMMAND HAS
REPORTED THAT ABOUT 50 PEASANTS OF VARIOUS AGES HAVE BEEN BY
TERRORISTS OF THE FARABUNDO MARTI NATIONAL LIBERATION FRONT
KIDNAPPED)>
(?38
```

```
((KNOWN-ENTITY ?4
        PEOPLE
        (SOCIAL-ROLE-OF ?4 MILITARY)
        (DESCRIPTION-OF ?4
                "THE ARCE BATTALION COMMAND")
        (DET ?4 "THE"))
 (KNOWN-EVENT ?38
        COMMUNICATION
        (AGENT-OF ?38 ?4)
        (PERFECT-TENSE ?38 "HAS")
        (PRESENT-TENSE ?38)
        (OBJECT-OF ?38 ?35)
        (PAST-TENSE ?38))
 (KNOWN-EVENT ?35
        KIDNAPPING
        (OBJECT-OF ?35 ?13)
        (AGENT-OF ?35 ?29)
        (PERFECT-TENSE ?35 "HAVE")
        (PRESENT-TENSE ?35)
        (PASSIVE ?35))
 (KNOWN-ENTITY ?13
        PERSON
        (PP-MODIFIER ?13 ?12 "OF")
        (SOCIAL-ROLE-OF ?13 CIVILIAN)
        (NUMBER-OF ?13 50)
        (DESCRIPTION-OF ?13 "ABOUT 50 PEASANTS"))
 (KNOWN-SOA ?12
       STATE-OF-AFFAIRS
       (NUMBER-OF ?12 PLURAL)
       (DESCRIPTION-OF ?12 "VARIOUS AGES"))
 (KNOWN-ENTITY ?23
        ORGANIZATION
        (NAME-OF ?23 "FMLN")
        (NAME-OF ?23
             "FARABUNDO MARTI NATIONAL LIBERATION FRONT")
        (SOCIAL-ROLE-OF ?23 TERRORISM)
        (DESCRIPTION-OF ?23
                "THE FARABUNDO MARTI NATIONAL LIBERATION FRONT")
        (DET ?23 "THE"))
 (KNOWN-ENTITY ?29
        PERSON
        (PP-MODIFIER ?29 ?23 "OF")
        (SOCIAL-ROLE-OF ?29 TERRORISM)
        (NUMBER-OF ?29 PLURAL)
        (DESCRIPTION-OF ?29 "TERRORISTS")))
NIL)
```

A.5 Output template

0. MESSAGE ID DEV-MUC3-0001

1. TEMPLATE ID	1
2. DATE OF INCIDENT	30 DEC 89
3. TYPE OF INCIDENT	KIDNAPPING
4. CATEGORY OF INCIDENT	TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)	"TERRORISTS"
6. PERPETRATOR: ID OF ORG(S)	"THE FARABUNDO MARTI NATIONAL LIBERATION FRONT"
7. PERPETRATOR: CONFIDENCE	REPORTED AS FACT
8. PHYSICAL TARGET: ID(S)	*
9. PHYSICAL TARGET: TOTAL NUM	*
10. PHYSICAL TARGET: TYPE(S)	*
11. HUMAN TARGET: ID(S)	"ABOUT 50 PEASANTS"
12. HUMAN TARGET: TOTAL NUM	50
13. HUMAN TARGET: TYPE(S)	CIVILIAN
14. TARGET: FOREIGN NATION(S)	-
15. INSTRUMENT: TYPE(S)	*
16. LOCATION OF INCIDENT	EL SALVADOR: SAN SALVADOR (CITY)
17. EFFECT ON PHYSICAL TARGET(S)	*
18. EFFECT ON HUMAN TARGET(S)	-

A.6 Correct key template

0. MESSAGE ID	DEV-MUC3-0001 (NOSC)
1. TEMPLATE ID	1
2. DATE OF INCIDENT	30 DEC 89
3. TYPE OF INCIDENT	KIDNAPPING
4. CATEGORY OF INCIDENT	TERRORIST ACT
5. PERPETRATOR: ID OF INDIV(S)	"TERRORISTS"
6. PERPETRATOR: ID OF ORG(S)	"FARABUNDO MARTI LIBERATION FRONT" / "FMLN"
7. PERPETRATOR: CONFIDENCE	REPORTED AS FACT
8. PHYSICAL TARGET: ID(S)	*
9. PHYSICAL TARGET: TOTAL NUM	*
10. PHYSICAL TARGET: TYPE(S)	*
11. HUMAN TARGET: ID(S)	"PEASANTS"
12. HUMAN TARGET: TOTAL NUM	50
13. HUMAN TARGET: TYPE(S)	CIVILIAN
14. TARGET: FOREIGN NATION(S)	-
15. INSTRUMENT: TYPE(S)	*
16. LOCATION OF INCIDENT	EL SALVADOR: SAN MIGUEL (DEPARTMENT): SAN LUIS DE LA REINA (TOWN)
17. EFFECT ON PHYSICAL TARGET(S)	*
18. EFFECT ON HUMAN TARGET(S)	-

DISTRIBUTION LIST

addresses	number of copies
RL/COES ATTN: Douglas White Griffiss AFB NY 13441-5700	10
Dr. Ralph Weischedel BBN Systems and Techniques 10 Moulton Street Cambridge MA 02138	5
RL/DOVL Technical Library Griffiss AFB NY 13441-5700	1
Administrator Defense Technical Info Center DTIC-FDAC Cameron Station Building 5 Alexandria VA 22304-6145	2
Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington VA 22209-2308	2
RL/COAC Griffiss AFB NY 13441-5700	1
HQ USAF/SCTT Washington DC 20330-5190	1
SAF/AQSC Pentagon Rm 4D 269 Wash DC 20330	1

Naval Warfare Assessment Center 1
GIDEP Operations Center/Code 30G
ATTN: E Richards
Corona CA 91720

HQ AFSC/XTH 1
Andrews AFB MD 20334-5000

HQ SAC/SCPT 2
OFFUTT AFB NE 68046

DTESA/RQE 1
ATTN: Mr. Larry G.McManus
Kirtland AFB NM 87117-5000

HQ TAC/DRIY 1
ATTN: Maj. Divine
Langley AFB VA 23665-5575

HQ TAC/DOA 1
Langley AFB VA 23665-5554

ASD/ENEMS 1
Wright-Patterson AFB OH 45433-6503

SM-ALC/MACEA 1
ATTN: Danny McClure
Bldg 237, MASOF
McClellan AFB CA 95652

WRDC/AAAI-4 1
Wright-Patterson AFB OH 45433-6543

WRDC/AAAI-2 1
ATTN: Mr Franklin Hutson
WPAFB OH 45433-6543

AFIT/LDEE 1
Building 642, Area B
Wright-Patterson AFB OH 45433-6583

WRDC/MTEL 1
Wright-Patterson AFB OH 45433

AAMRL/HE 1
Wright-Patterson AFB OH 45433-6573

Air Force Human Resources Lab 1
Technical Documents Center
AFHRL/LRS-TDC
Wright-Patterson AFB OH 45433

AUL/LSE 1
Bldg 1405
Maxwell AFB AL 36112-5564

HQ AFSPACECOM/XRA 1
STINFO Officer
ATTN: Dr. W. R. Matoush
Peterson AFB CO 80914-5001

HQ ATC/TTOI 1
ATTN: Lt Col Killian
Randolph AFB TX 78150-5001

AFLMC/LGY 1
ATTN: Maj. Shaffer
Building 205
Gunter AFS AL 36114-6693

US Army Strategic Def 1
CSSD-IM-PA
PO Box 1500
Huntsville AL 35807-3801

Ofc of the Chief of Naval Operation 1
ATTN: William J.Cook
Navy Electromagnetic Spectrum Mgt
Room 5A678, Pentagon (OP-941)
Wash DC 20350

Commanding Officer 1
Naval Avionics Center
Library D/765
Indianapolis IN 46219-2189

Commanding Officer 1
Naval Ocean Systems Center
Technical Library
Code 96428
San Diego CA 92152-5000

Cmdr 1
Naval Weapons Center
Technical Library/C3431
China Lake CA 93555-6001

Superintendent 1
Code 1424
Naval Postgraduate School
Monterey CA 93943-5000

Space & Naval Warfare Systems Comm 1
Washington DC 20363-5100

CDR, U.S. Army Missile Command 2
Redstone Scientific Info Center
AMSMI-RD-CS-R/ILL Documents
Redstone Arsenal AL 35898-5241

Advisory Group on Electron Devices 2
201 Varick Street, Rm 1140
New York NY 10014

Los Alamos National Laboratory
Report Library
MS 5000
Los Alamos NM 87544 — 1

AEDC Library
Tech Files/MS-100
Arnold AFB TN 37389 — 1

Commander, USAG
ASQH-PCA-CRL/Tech Lib
Bldg 61801
Ft Huachuca AZ 85613-6000 — 1

1839 EIG/EIT
Keesler AFB MS 39534-6348 — 1

AFEWC/ESRI
San Antonio TX 78243-5000 — 1

ESD/XRR
Hanscom AFB MA 01731-5000 — 1

ESD/SZM
Hanscom AFB MA 01731-5000 — 1

SEI JPO
ATTN: Major Charles J. Ryan
Carnegie Mellon University
Pittsburgh PA 15213-3890 — 1

Director NSA/CSS
T5122/TDL
ATTN: D W Marjarum
Fort Meade MD 20755-6000 — 1

Director NSA/CSS 1
W157
9800 Savage Road
Fort Meade MD 21055-6000

NSA 1
ATTN: D. Alley
Div X911
9800 Savage Road
Ft Meade MD 20755-6000

Director 1
NSA/CSS
W11 DEFSMAC
ATTN: Mr. Mark E. Clesh
Fort George G. Meade MD 20755-6000

Director 1
NSA/CSS R12
ATTN: Mr. Dennis Heinbuch
9800 Savage Road
Fort George G. Meade MD 20755-6000

DoD 1
R31
9800 Savage Road
Ft. Meade MD 20755-6000

DIRNSA 1
R509
9800 Savage Road
Ft Meade MD 20775

Director 1
NSA/CSS
R09
Fort George G. Meade MD 20755-6000

DOD Computer Center 1
C/TIC
9800 Savage Road
Fort George G. Meade MD 20755-6000

SUNY at Buffalo 1
Computer Science Department
Attn: Dr Stuart C. Shapiro
226 Bell Hall
Buffalo NY 14260

University of Pennsylvania
Attn: Dr Aravind Joshi
Dept of Comp and Info Science
Philadelphia PA 19104-6389 1

University of Pennsylvania
Attn: Dr Mitch Marcus
Dept of Comp and Info Sciences
Philadelphia PA 19104 1

DARPA/ISTO
Attn: Dr Charles Wayne
1400 Wilson Blvd
Arlington VA 20375 1

Language Systems, Inc.
Attn: Dr Christine Montgomery
6269 Variel Ave, Suite 200
Woodland Hills CA 91367 1

Calspan Corporation
Attn: Dr Jeanette Neal
P.O. Box 400
Buffalo NY 14225 1

USC/ISI
Attn: Dr Ed Hovy
4676 Admiralty Way
Marina Del Ray CA 90292 1

Naval Ocean Systems Center
Attn: Ms Beth Sundheim
Code 444
San Diego CA 92152 1

Courant Institute
New York University
Attn: Dr Ralph Grishman
251 Mercer Street
New York NY 10012 1

Dragon Systems, Inc.
Attn: Mr Jim Baker
90 Bridge St
Newton MA 02158 1

Cognitive Systems Inc. 1
Attn: Dr Anatole Gershman
234 Church St
New Haven CT 06510

Univerity of Pennsylvania 1
Attn: Dr Bonnie Webber
Dept of Comp & Info Sciences
Philadelphia PA 19104

AFOSR/NM 1
Attn: Prof Abraham Waksman
Bolling AFB DC 20332-6448

AFOSR/NM 1
Attn: Dr Charles J. Holland
Bolling AFB DC 20332-6448

ESD-MITRE Software Center Librar, 1
Attn: Ms J. A. Clapp
MITRE Corp D-70, MS A-359
Burlington Road
Bedford MA 01730

AFHRL/ID 1
Attn: James Parlett, Major, USAF
Brooks AFB TX 78235-5601

International Ctr for Machine 1
Translation
Carnegie-Mellon University
Attn: Prof Sergei Nirenburg
Pittsburgh PA 15213

University of Rochester 2
Chairman, Dept of Comp Science
Attn: Prof James F. Allen
Computer Studies Building
Rochester NY 14627

University of Massachusetts 2
Lederle Graduate Research Center
Attn: Dr Beverly Woolf
COINS Department
Amherst MA 01003-0001

Univ of Arkansas/Little Rock 1
Attn: Drs. Walter & Sally Sedelow
Computer Science Dept
P.O. Box 942
Heber Springs, Arkansas 72543

SRI International 1
ATTN: Douglas E. Appelt
AI Center for Study of Language
and Information
Menlo Park CA 94025

Columbia University 1
ATTN: Dr. Kathy McKeown
Dept Computer Science Bldg
New York, NY 10027

Univ of Deleware 1
ATTN: Dr. Katheleen McCoy
Dept of Computer & Info Sciences
Newark DE 19716

Dr. Donald Walker 1
Bellcore MTRE 2A379
445 South Street, Box 1910
Morristown NJ 07960

Harvard University 1
ATTN: Bill Woods
Aiken Computer Lab
Cambridge MA 02138

Intelligence Applications (USA), Inc 1
ATTN: Dr. Robert Milne
c/o Brattle Research Corp
55 Wheeler Street
Cambridge MA 02138

New Mexico State University 1
ATTN: Dr. Yorick Wilks
Computing Research Laboratory
Box 3001
Las Cruces NM 88003

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.